CS167: Machine Learning

Decision Tree

Thursday, February 29th, 2024

Drake University

Announcements

- Heads up: Quiz #1
  - due tonight by 11:59pm

- Notebook #3: Cross Validation
  - due tonight by 11:59pm
  - to submit, download the `ipynb` file from Colab

Review

- Evaluation Metrics
  - Classification metrics
  - Regression metrics

Review: classification metrics (accuracy)

- **Accuracy**: The fraction of test examples your model predicted correctly
  - *Example*: 17 out of 20 = 0.85 accuracy

- Issues with accuracy: suppose that a blood test for cancer has 99% accuracy

  - *can we safely assume this is a really good test?*
    - If the dataset is *unbalanced*, accuracy is not a reliable metric for the real performance of a classifier because it will yield misleading results
    - **Example**: Most people don't have cancer

  - Beware of what your metrics don't tell you

Review: classification metrics (`confusion matrix`)

- **Confusion matrix**: A specific table layout that allows the visualization of the performance of an algorithm.
- Each row represents instances in an actual class
- While each column represents the instances in a predicted class
  - It makes it easy to see where your model is confusing the predicted and actual results. For a binary classification problem:

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Review: example (`confusion matrix`)

- **Confusion matrix:** A specific table layout that allows the visualization of the performance of an algorithm

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0
Result	FN	FN	TP	TP	TP	TP	TP	TP	FP	TN	TN	TN

- Given the following confusion matrix:
  - how many true positive? 6
  - how many true negatives? 3
  - how many false positive? 1
  - how many false negatives? 2

		Predicted class	
		P	*N*
Actual Class	*P*	6	2
	N	1	3

Review: regression metrics (MAE vs. MSE)

- **Mean Absolute Error (MAE):** the average difference (absolute difference ie, always a positive value) between the actual and predicted target values

$$\frac{\sum_{\text{test example } x_i} |actual(x_i) - predicted(x_i)|}{\text{number of test examples}}$$

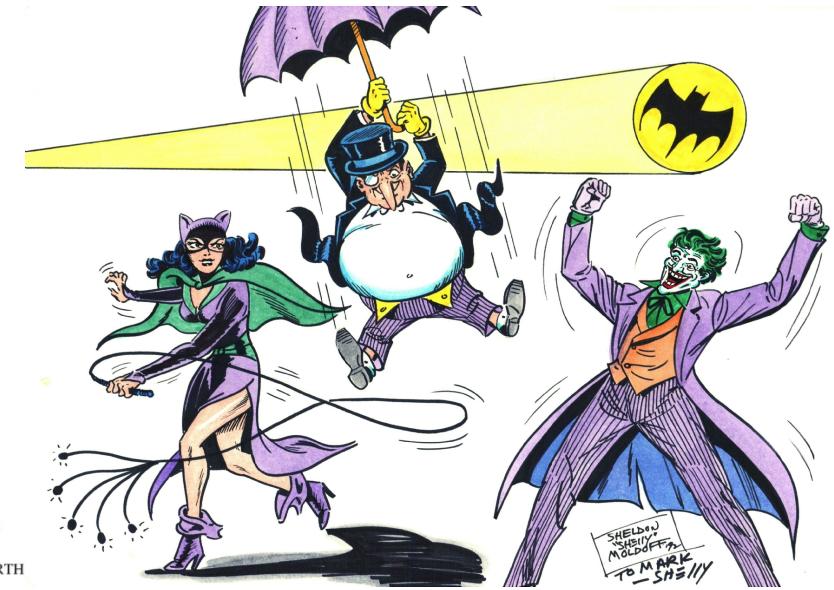
- **Mean Squared Error (MSE):** the average squared difference between the actual and predicted targets

$$\frac{\sum_{\text{test example } x_i} (actual(x_i) - predicted(x_i))^2}{\text{number of test examples}}$$

Today's Agenda

- Decision Tree
- Entropy

Decision: Are these comic book characters good or evil?



Problem: Is your date good or bad?

	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

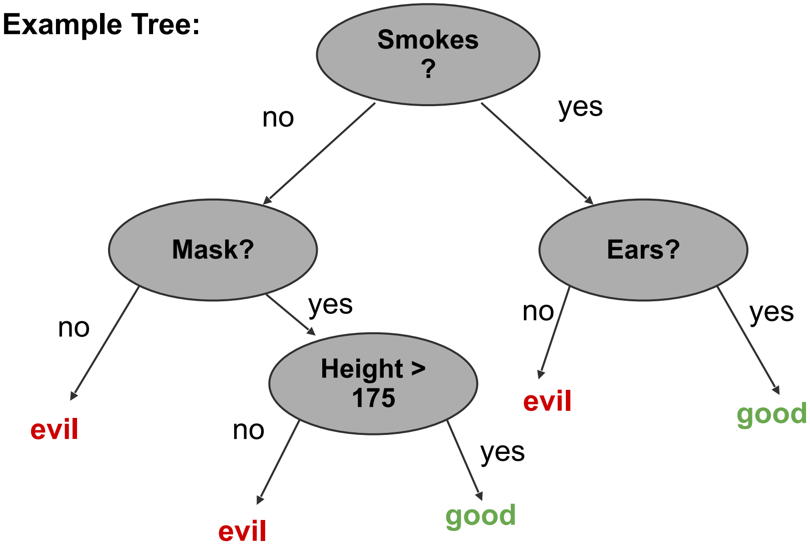
Test Data

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	?

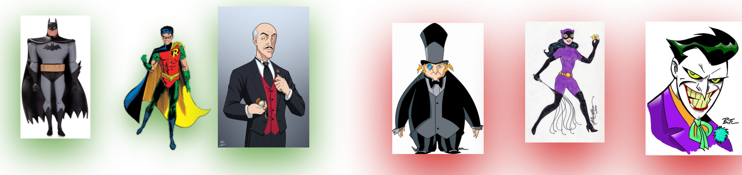
Example Tree:



Question: Is this a good tree?

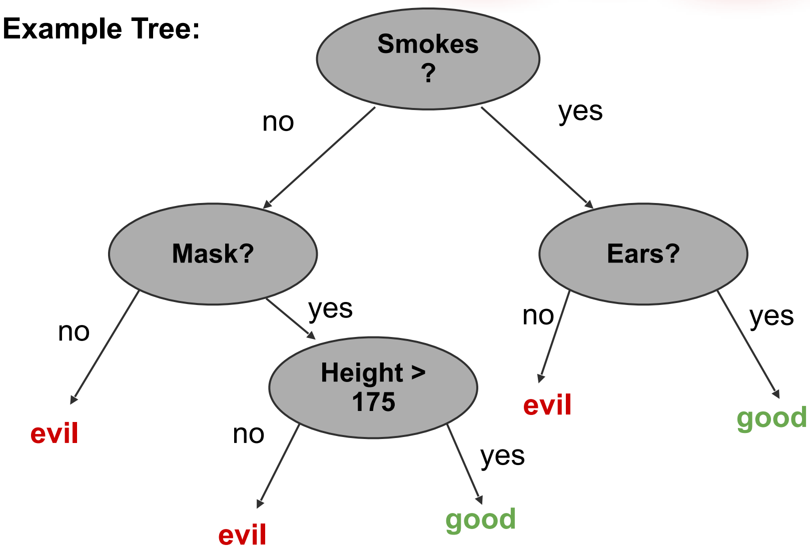
Is this tree **consistent:** would it classify everyone correctly?

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?

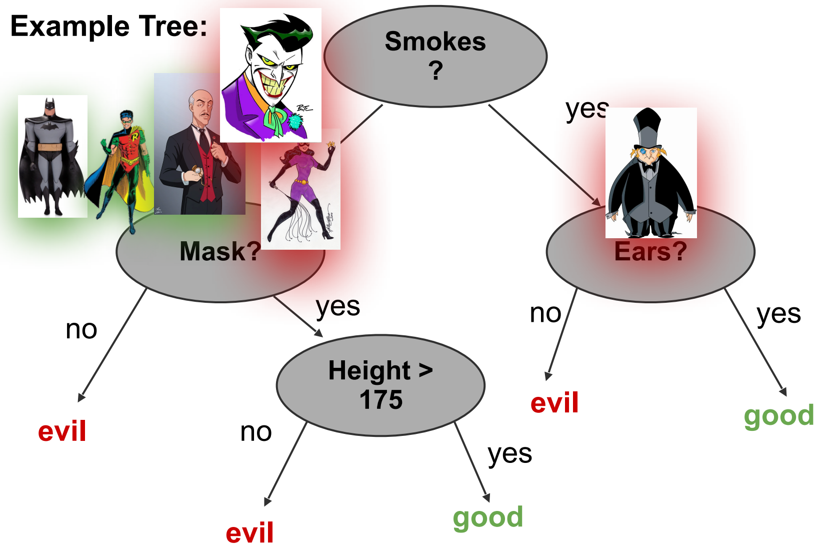
Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

Let's classify the characters based on the value of attribute 'Smokes'

	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?

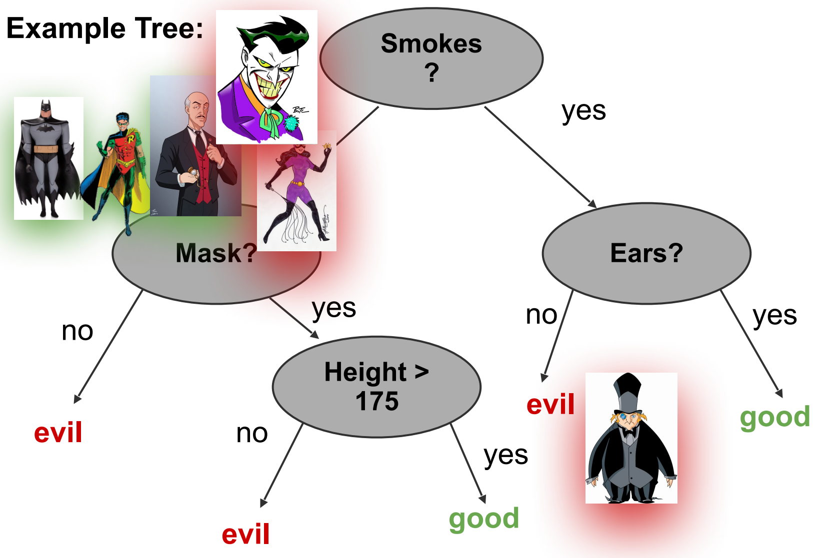
Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

Let’s classify the characters based on the value of attribute ‘Ears’

	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?

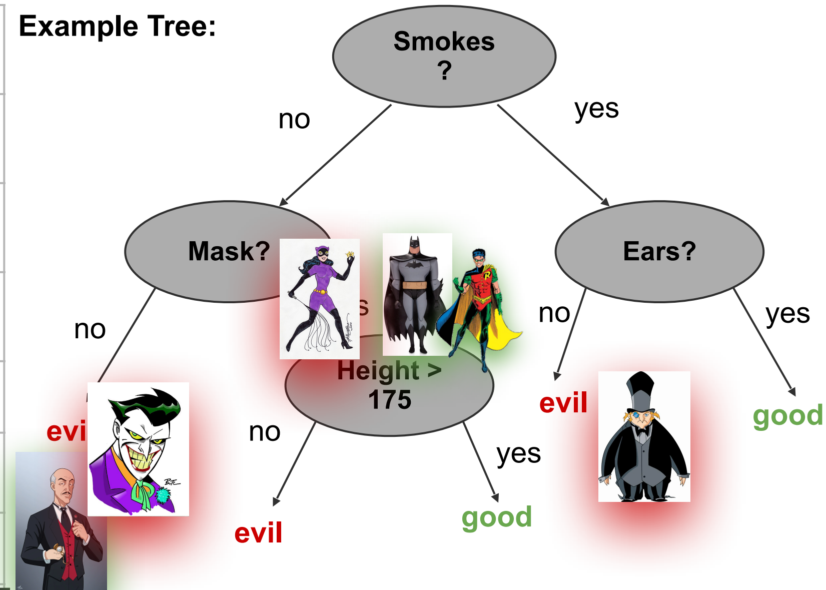
Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

Let's classify the characters based on the value of attribute 'Mask'

	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?

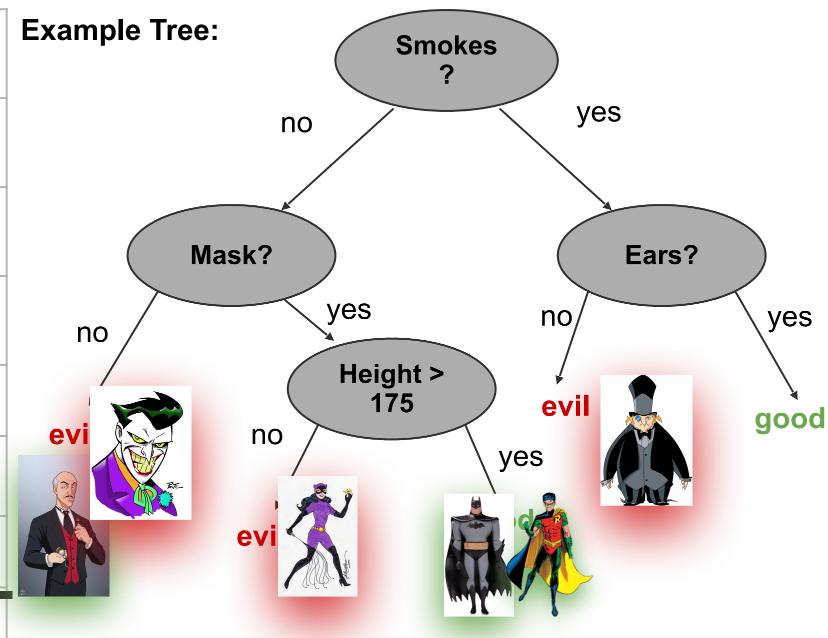
Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

Let's classify the characters based on the value of attribute 'Height'

	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:



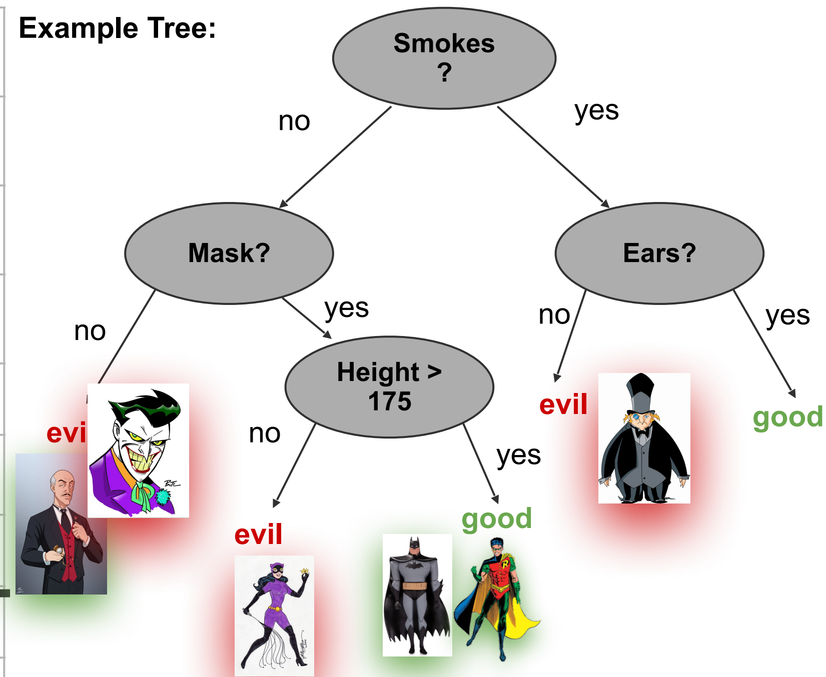
Answer:

- 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:

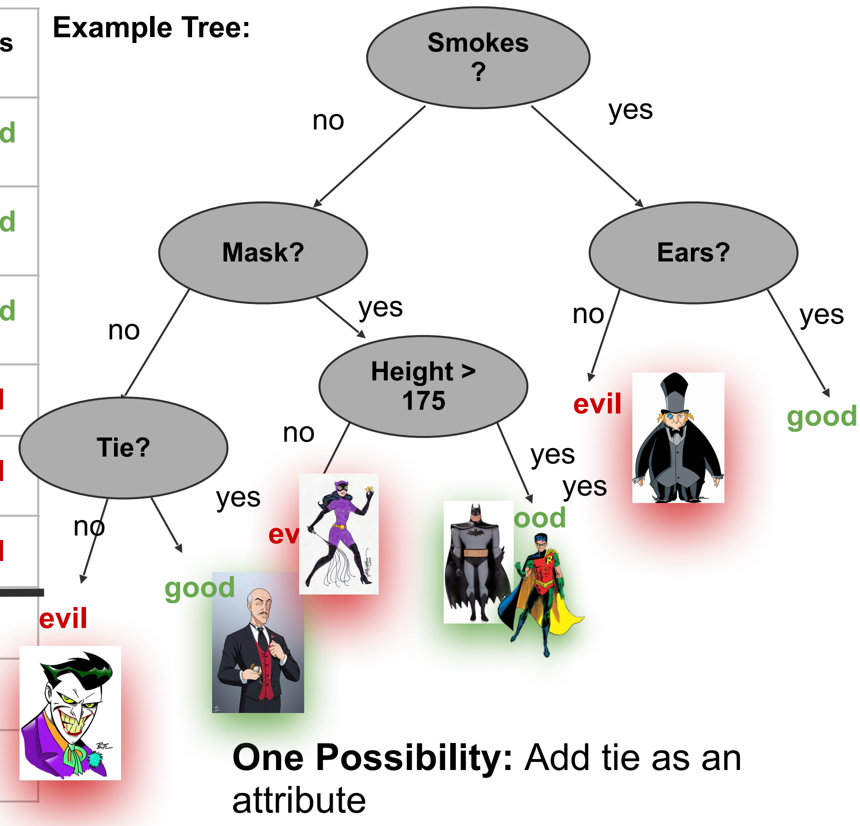


Question: What can we do to make this tree consistent?

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:

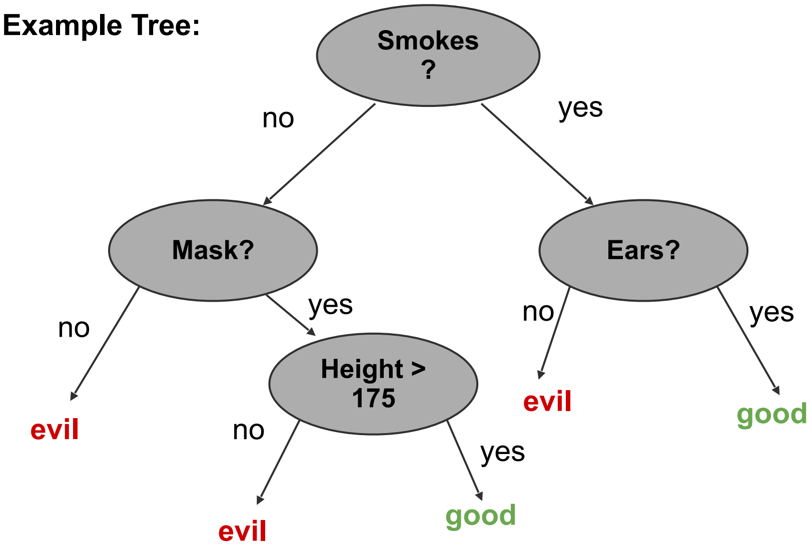


One Possibility: Add tie as an attribute

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	?

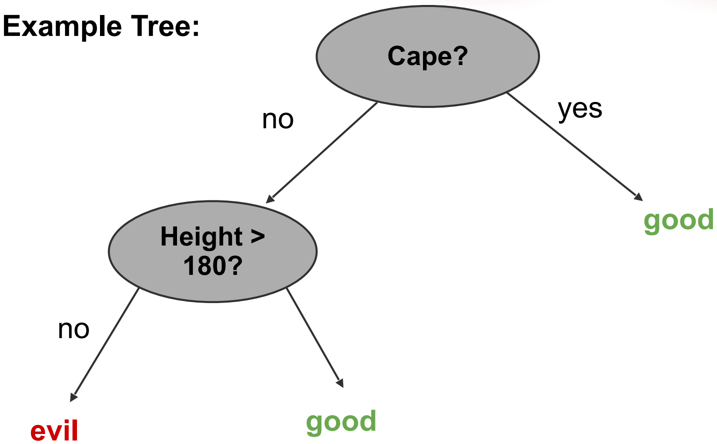
Example Tree:



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	?

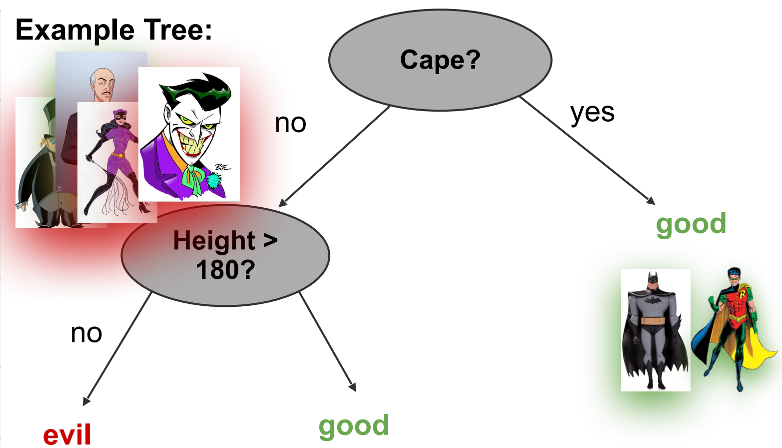
Example Tree:



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	?

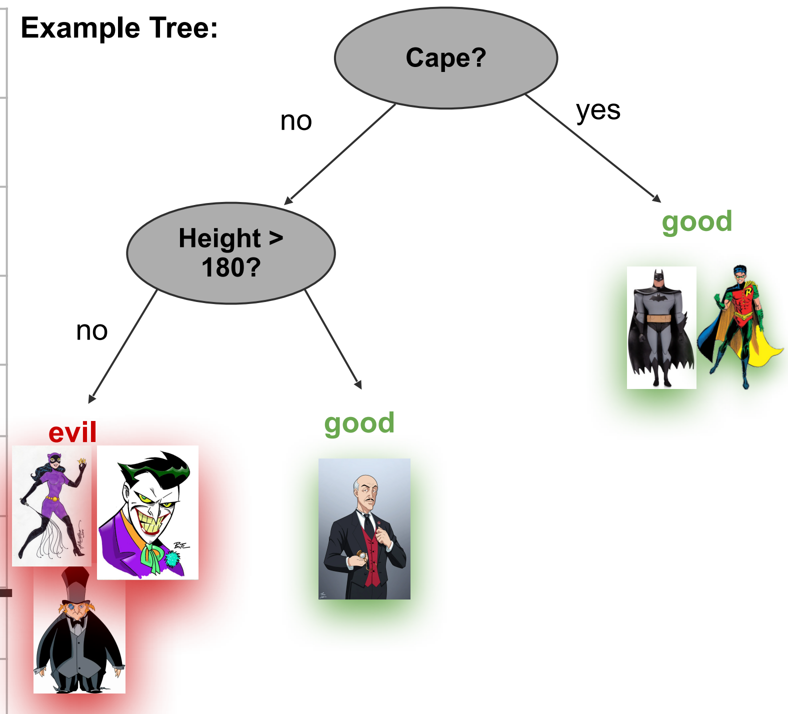
Example Tree:



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	?

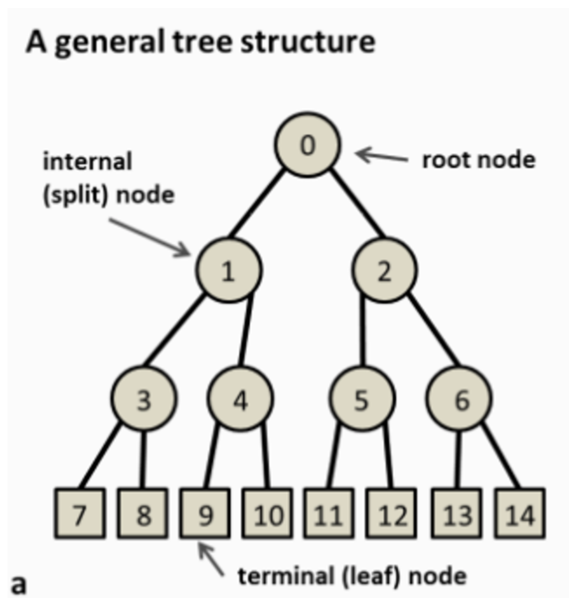
Example Tree:



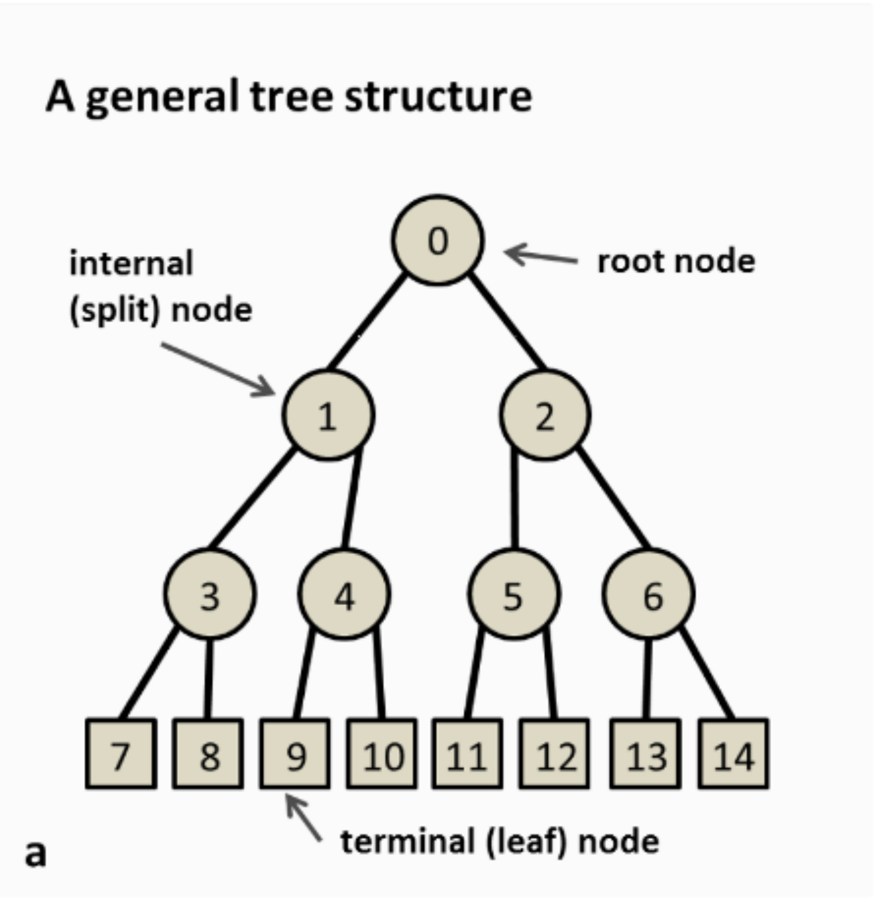
Decision Tree

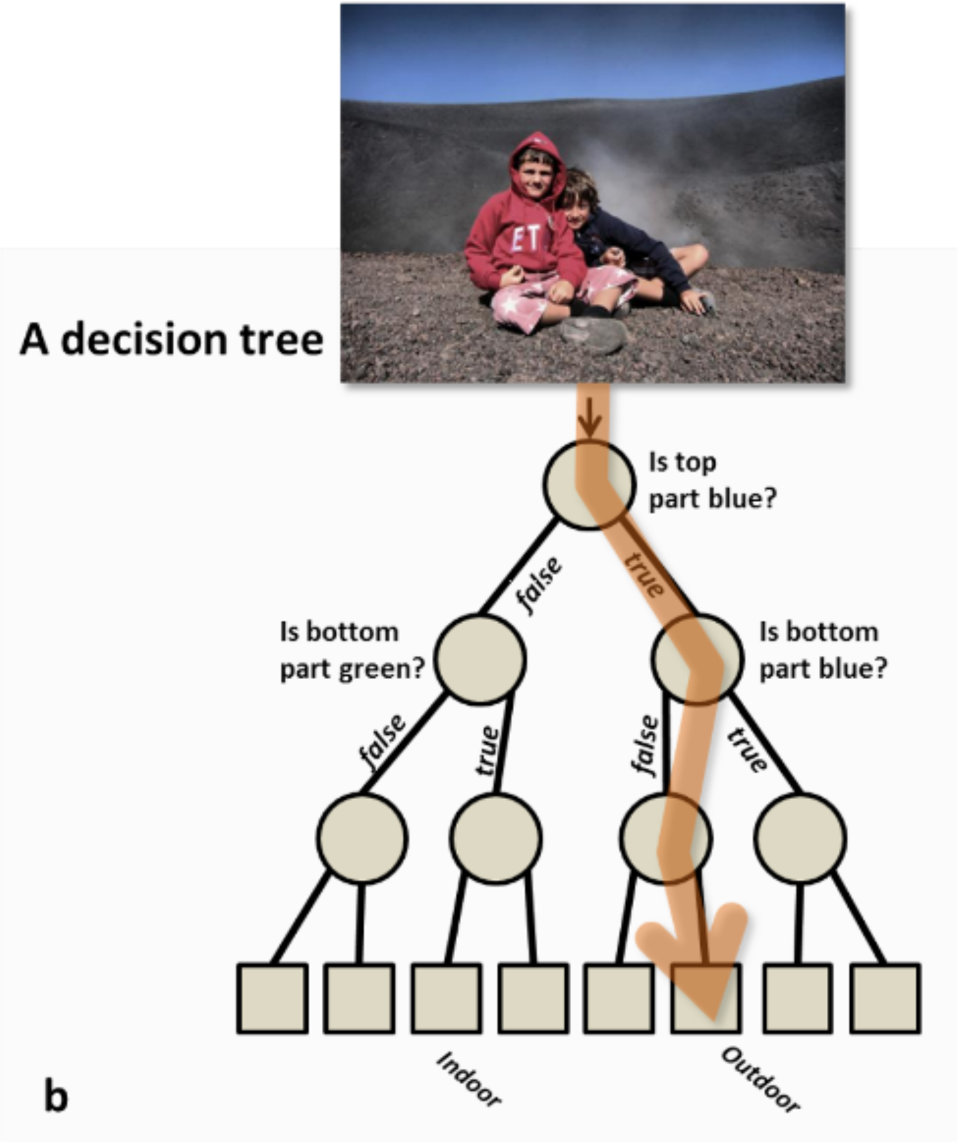
Tree Data Structure

- **Tree:** a common data structure that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node, represented as a set of liked nodes.



Decision Tree





Today's Agenda

- Decision Tree
- Entropy

Decision Tree: Another Example

Features or Attributes:

Target feature is whether or not a person will stay at a restaurant (T, F) with the following predictor 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).

Decision Tree: Another Example

Restaurant dataset

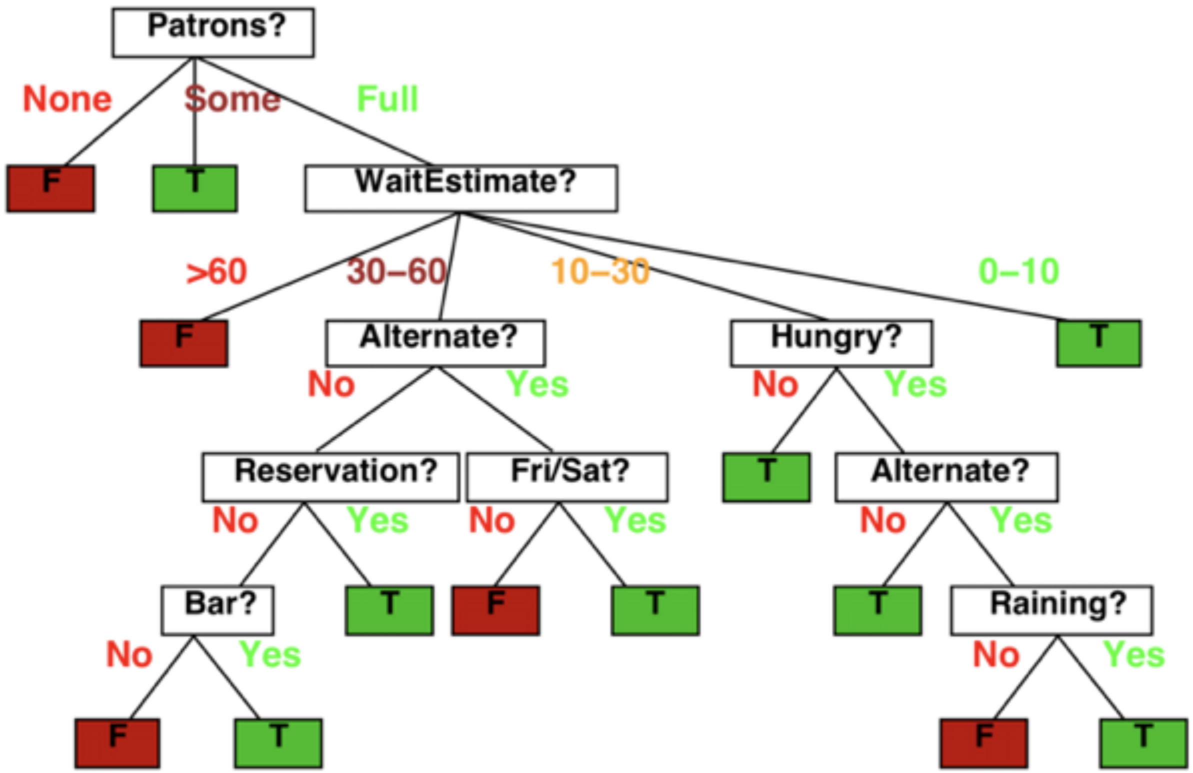
Predictor feature

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

Target feature

Decision Tree: Another Example

Example Tree



Decision Tree: Another Example

Example Tree

Predictor feature											Target feature
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

Consistent and Generalize

- Is this tree consistent with the training examples?
  - do all of the training examples get categorized appropriately?

- Will this tree generalize well to new examples?
  - how well will new examples (test set) perform?

Growing or Building a Decision Tree

- Great, now how do I build (grow) a tree?
  - One algorithm that builds a decision tree is called:
    - **ID3 Decision Tree Learning Algorithm**

ID3 Decision Tree Learning (Main Loop)

1. A ← select 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?

Choosing a feature

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

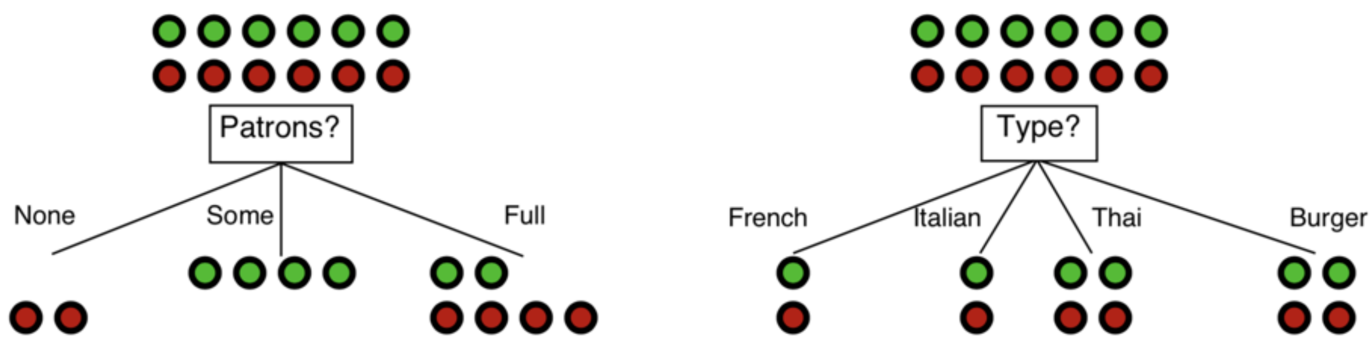
Predictor feature											
Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	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

Target feature

Choosing a feature

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

Red = false target value, Green = true target value

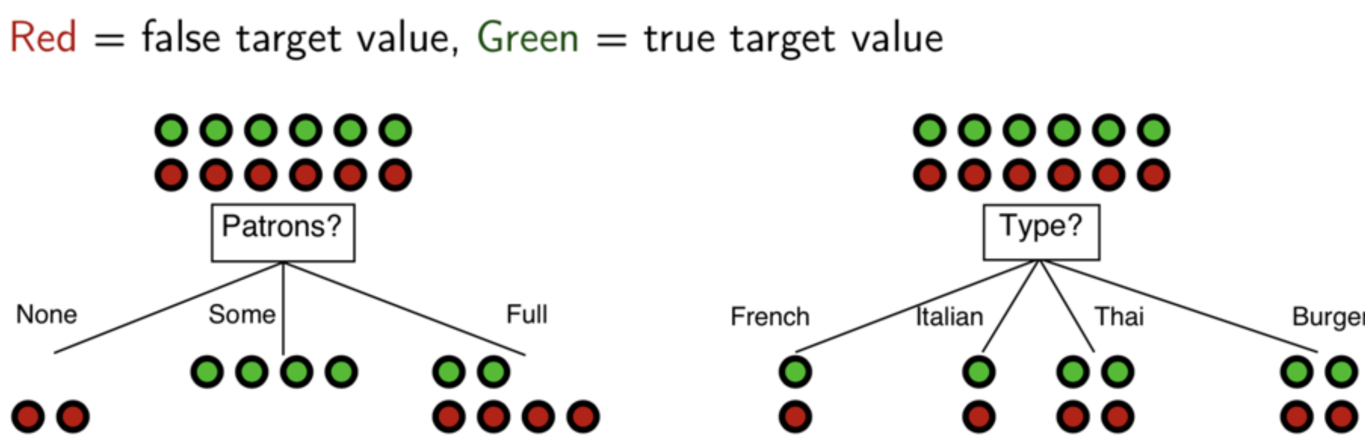


- *Wait column is the target value*

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*

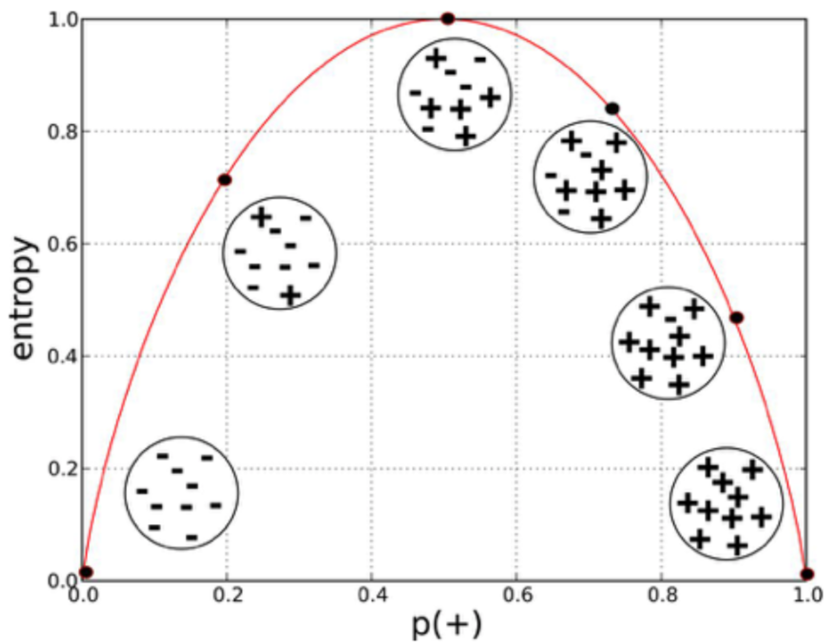


- *Wait column is the target value*

Decision Tree: Entropy

Entropy is a measure of impurity/randomness

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



Decision Tree: Calculating Entropy Prior

Prior Probability: aka the 'prior'

- the split of the examples
- Out of 14 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 the entropy when prior is $\langle P_1, \ldots, P_c \rangle$ is:

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

Decision Tree: Calculating Entropy Prior

Calculating the entropy when prior is $\langle P_1,\ldots,P_c \rangle$ is:

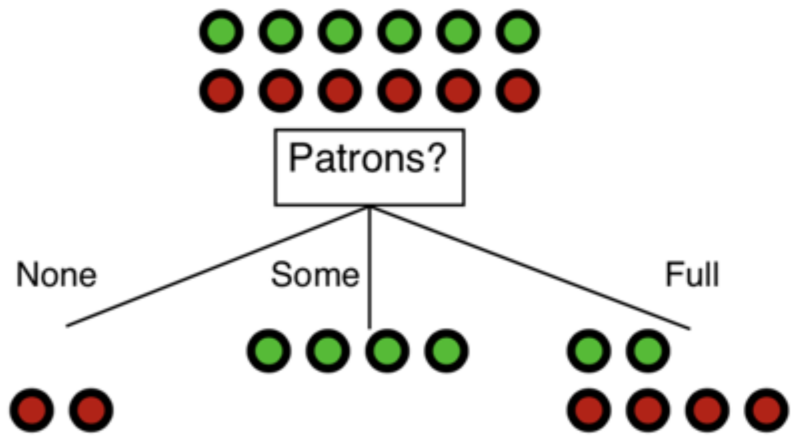
$$Entropy(\langle P_1, \ldots, P_c \rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

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

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

$$\log_2 (3) = 1.584962501$$

Decision Tree: Entropy Calculation Example



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

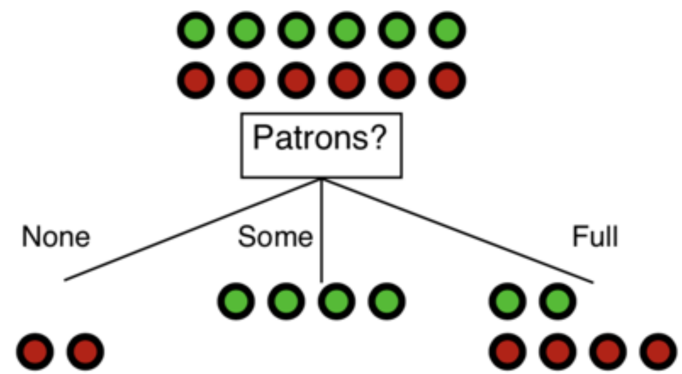
$$-\frac{0}{2}\log_2\frac{0}{2}-\frac{2}{2}\log_2\frac{2}{2}=0,$$

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

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

Decision Tree: Entropy Calculation Exercise

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.



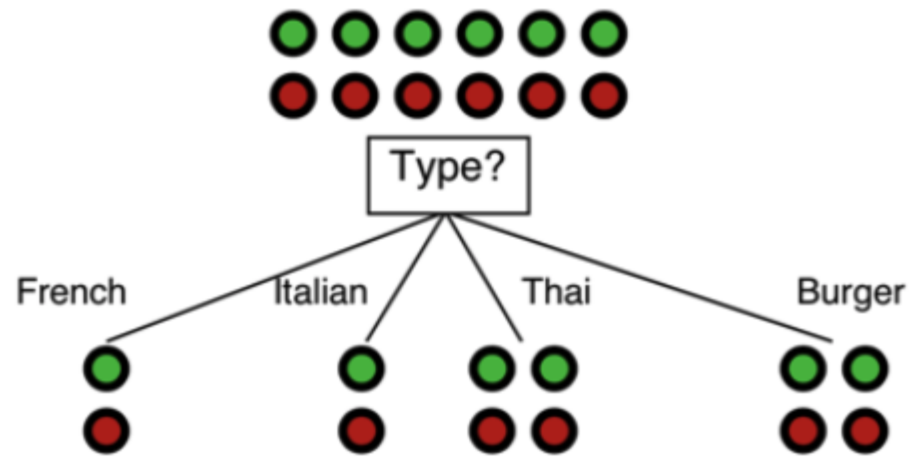
Then, the *expected entropy* remaining after testing the *Patrons* is

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

Decision Tree: Entropy Calculation Example

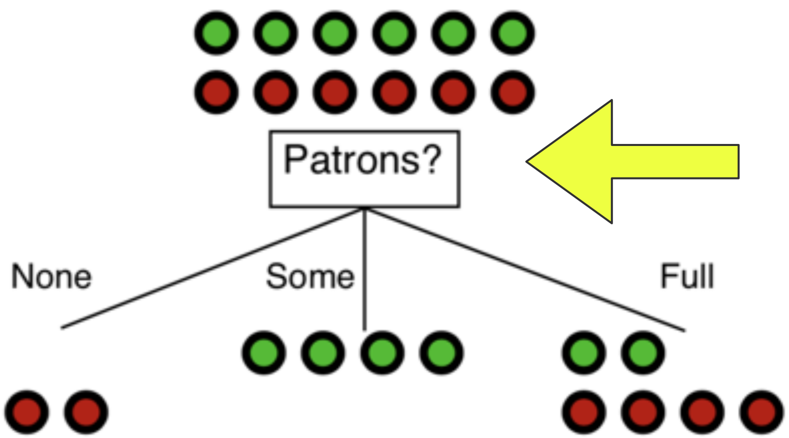
What is the expected entropy for Type?

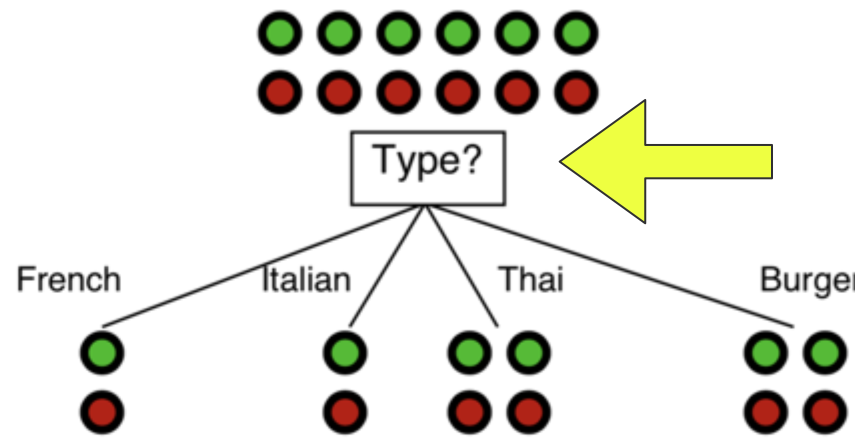
Can you say without doing the math?



Decision Tree: Information Gain Example

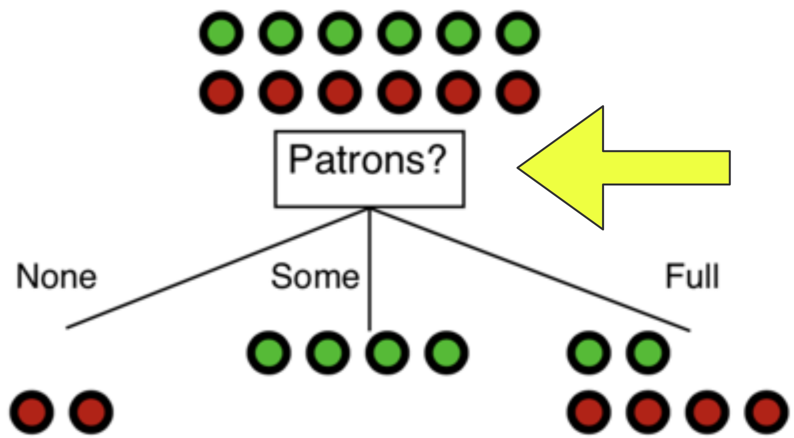
- We need to calculate entropy for each feature (eg, *Patron*, *Type*) as a candidate





Decision Tree: Information Gain Example

- Calculate entropy for feature *Patron* as a candidate

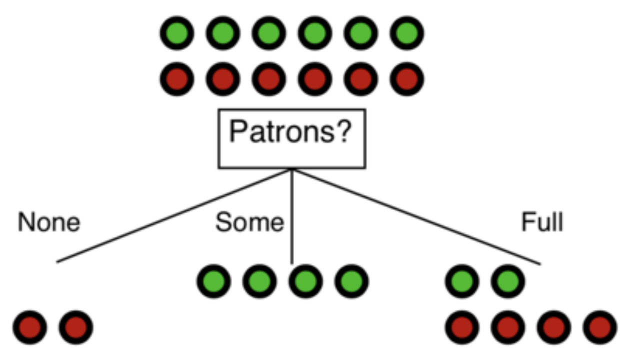


Decision Tree: Information Gain Example

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

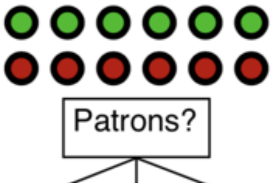
InformationGain() = Entropy (before) - Expected Entropy (after)

InformationGain(Patrons) = 1.0 - 0.459 = 0.541



Decision Tree: Calculating Information Gain

- **<u>Step 1:</u>** Calculate the entropy of the distribution of the classes before the node you are testing.
    - this is the **entropy_before**



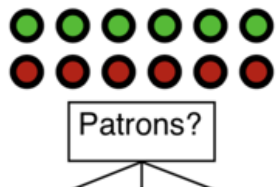
- entropy of prior $\langle 0.5, 0.5 \rangle$
    - $-0.5 \log_2 (0.5) - 0.5 \log_2 (0.5) = 1$

So, Entropy_before = 1

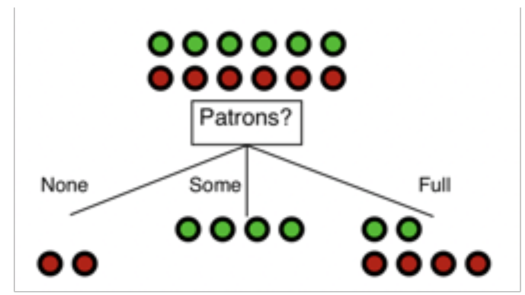
Decision Tree: Calculating Information Gain

- **<u>Step 1:</u>** Calculate the entropy of the distribution of the classes before the node you are testing. This is the **entropy before**

Entropy_before = 1



- **<u>Step 2:</u>** Calculate the **expected entropy**
  - The weighted sum of the entropy of each split of the data



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

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

Decision Tree: Calculating Information Gain

- Step 1: Calculate the entropy of the distribution of the classes before the node you are testing. This is the **entropy before**

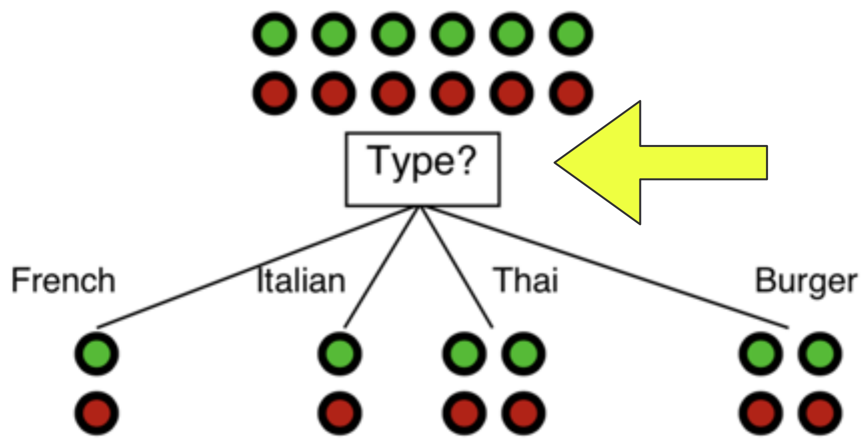
- Step 2: Calculate the **expected entropy**
  - The weighted sum of the entropy of each split of the data

- Step 3: Find the difference between the **entropy before** and **expected entropy**
  - **Information Gain(Patron)** = **Entropy_before(Patron)** - **Expected_entropy(Patron)**

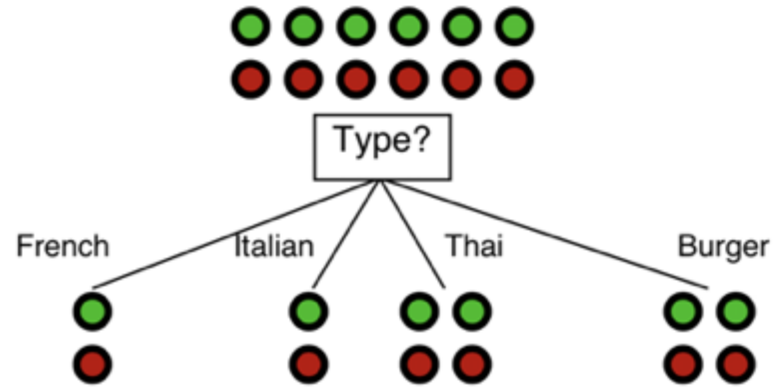
    = 1 - 0.459
    = 0.541

Decision Tree: Entropy Calculation Example

- Calculate entropy for feature *Type* as a candidate



Decision Tree: Calculating Information Gain



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

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

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

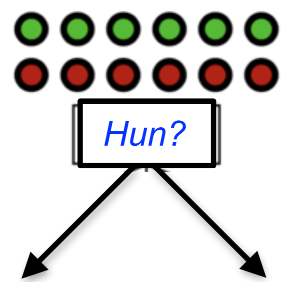
$$= \frac{2}{12} \cdot 1 + \frac{2}{12} \cdot 1 + \frac{4}{12} \cdot 1 + \frac{4}{12} \cdot 1 = 1$$

So,

$$Gain(Type) = 1 - 1 = 0$$

Decision Tree: Exercise Information Gain

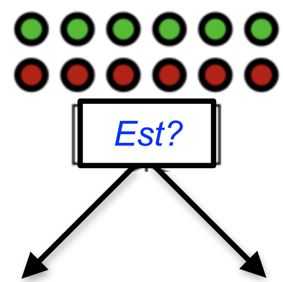
- Calculate the Information Gain for *Hun*:



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

Decision Tree: Exercise Information Gain

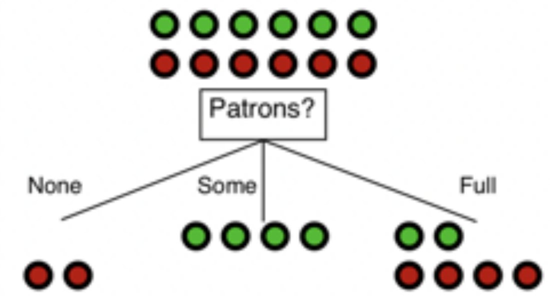
- Calculate the Information Gain for *Hun*:



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

Decision Tree: Exercise Information Gain

- 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(Patrons) = 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



Decision Tree: What to do with numeric features?

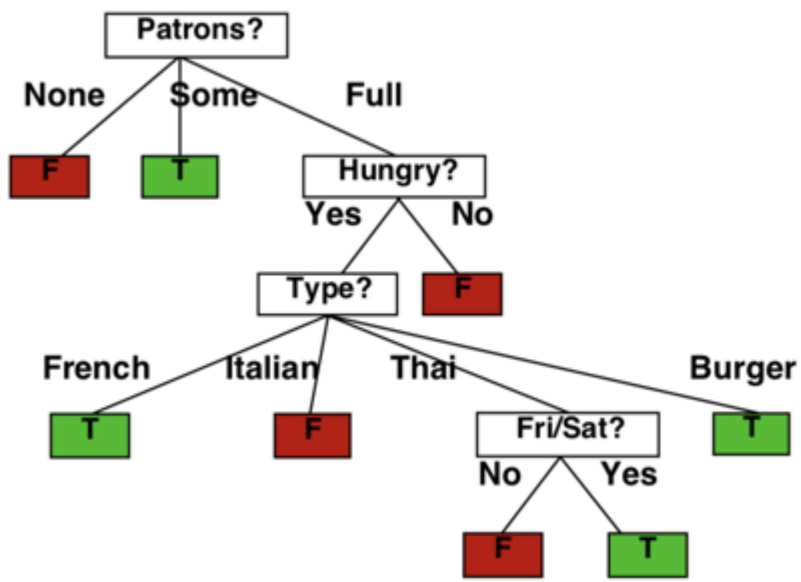
What do we do if we have numeric (even continuous-valued) features like age from the titanic dataset or petal length from the iris dataset?

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.

Decision Tree Size Discussion

Decision tree learned from the 12 examples:



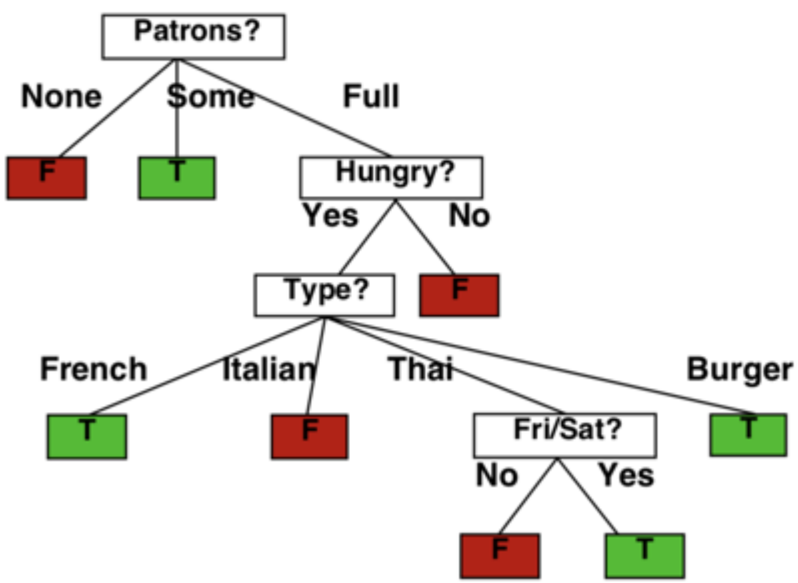
Decision Tree Size Discussion

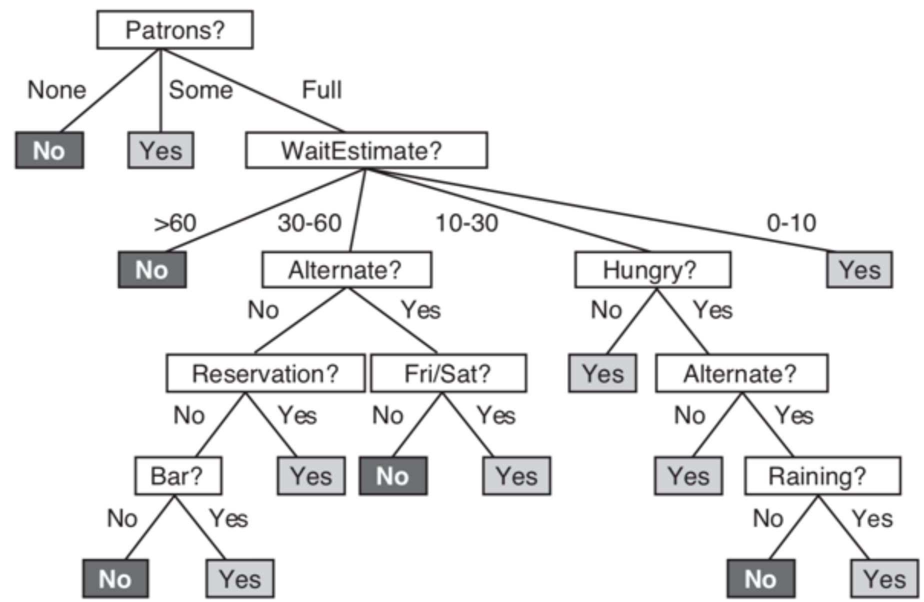
Many different consistent trees possible:

What quality is preferably?

More nodes v fewer nodes?

What are the consequences of having a deep tree with many nodes?





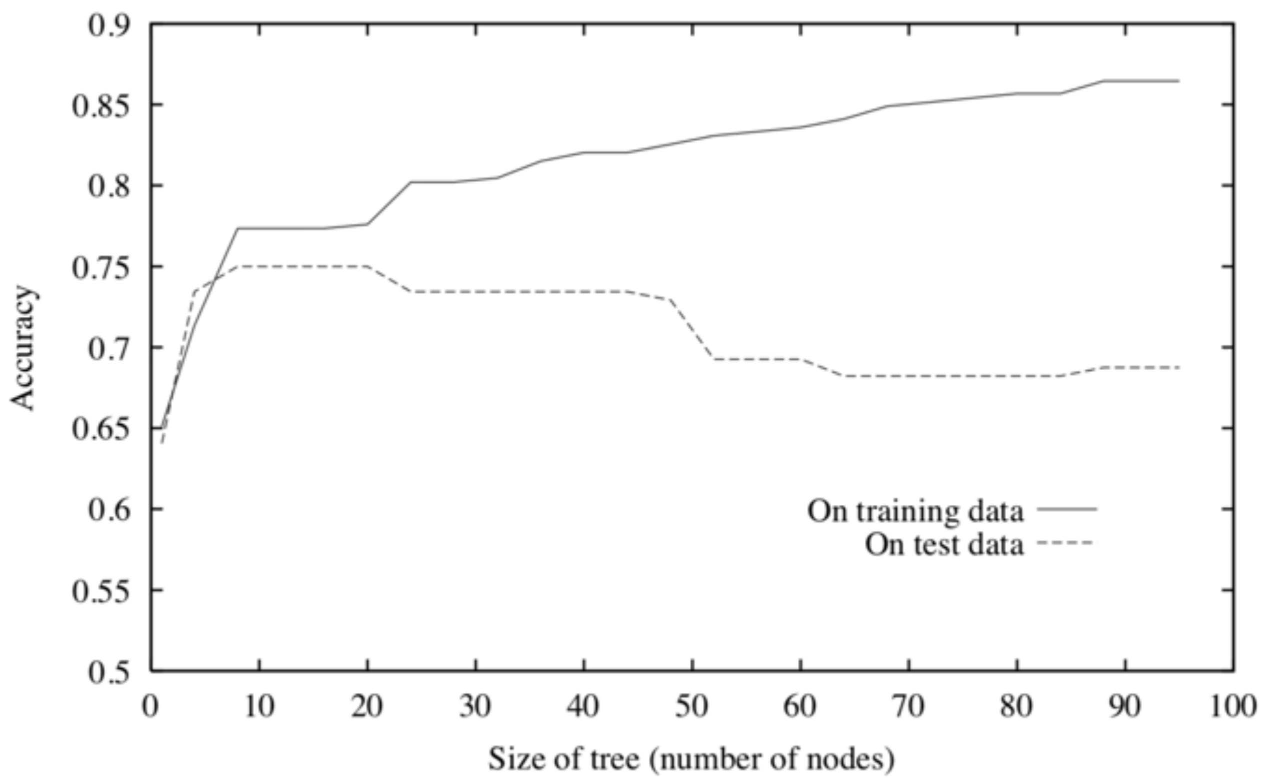
Inductive Bias of ID3 Algorithm

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

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

Overfitting

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



Avoid 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

Avoid Overfitting

- 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