Decision tree

Ngô Minh Nhựt

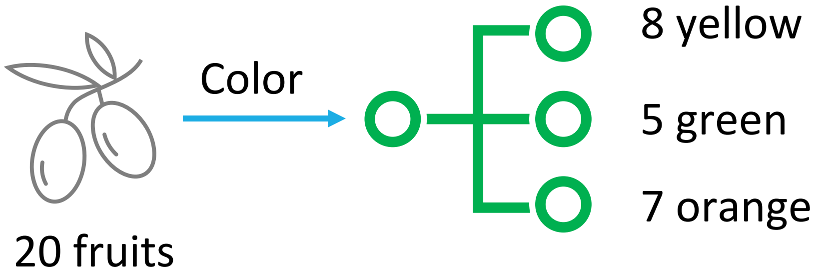
2025

Outline

- Classification idea
- Learn decision trees
  - Entropy
  - Information gain
  - Decision tree construction algorithm
- Overfitting and pruning

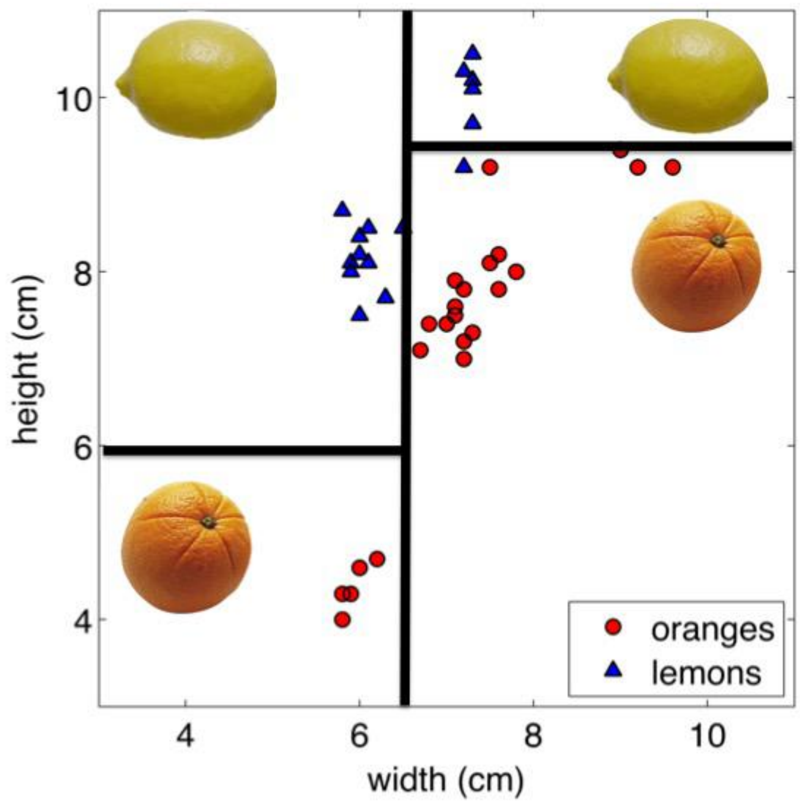
Classification idea

- Pick an attribute, do a simple test
- Conditioned on a choice, pick another attribute, do another test
- In the leaves, assign a class with majority vote
- Do other branches as well



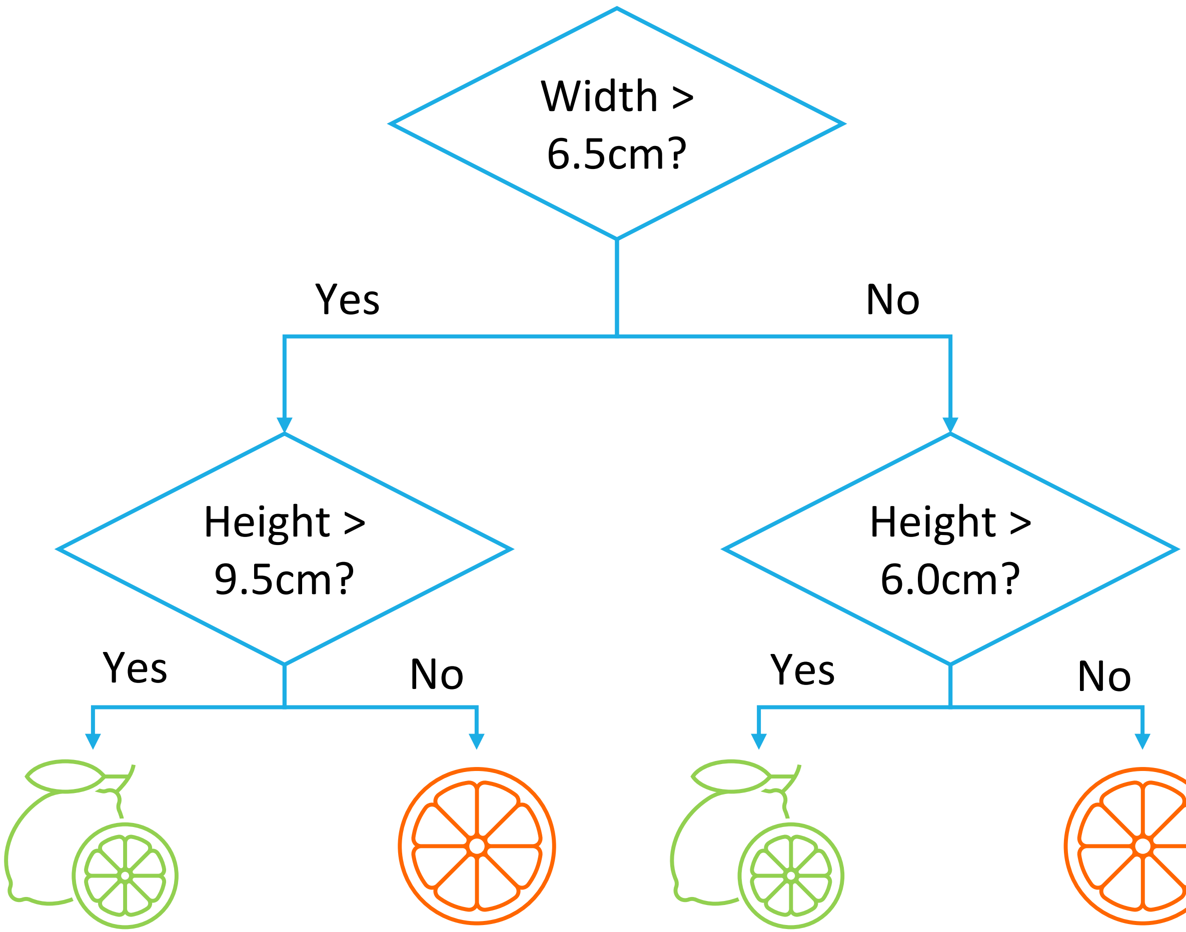
Classification idea

- Decision boundaries



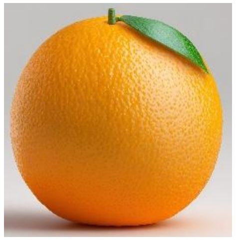
Source: Zemel

Classification idea



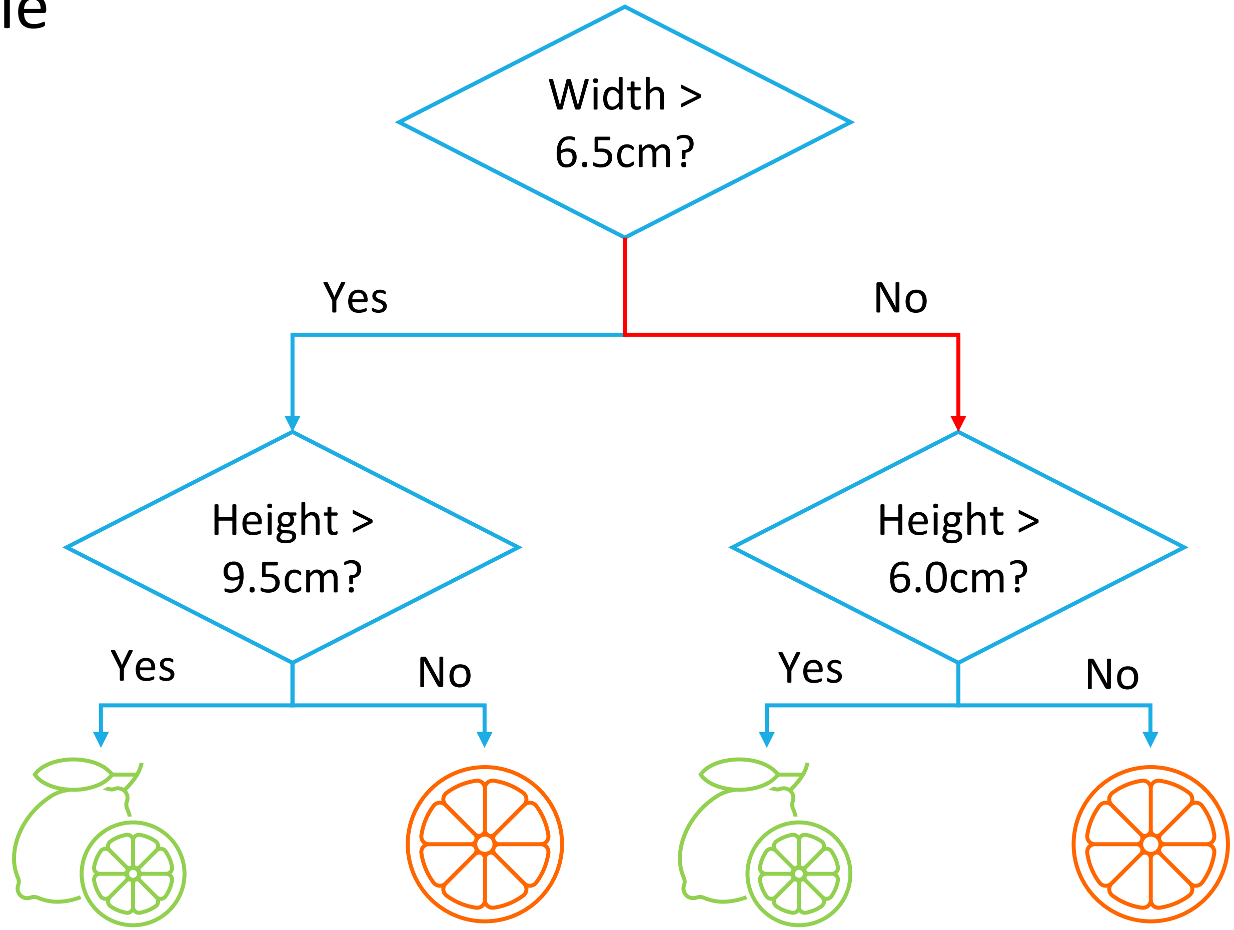
Classification idea

- Test example



5 cm

5 cm

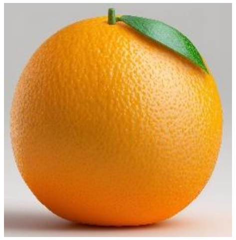


Source: Internet

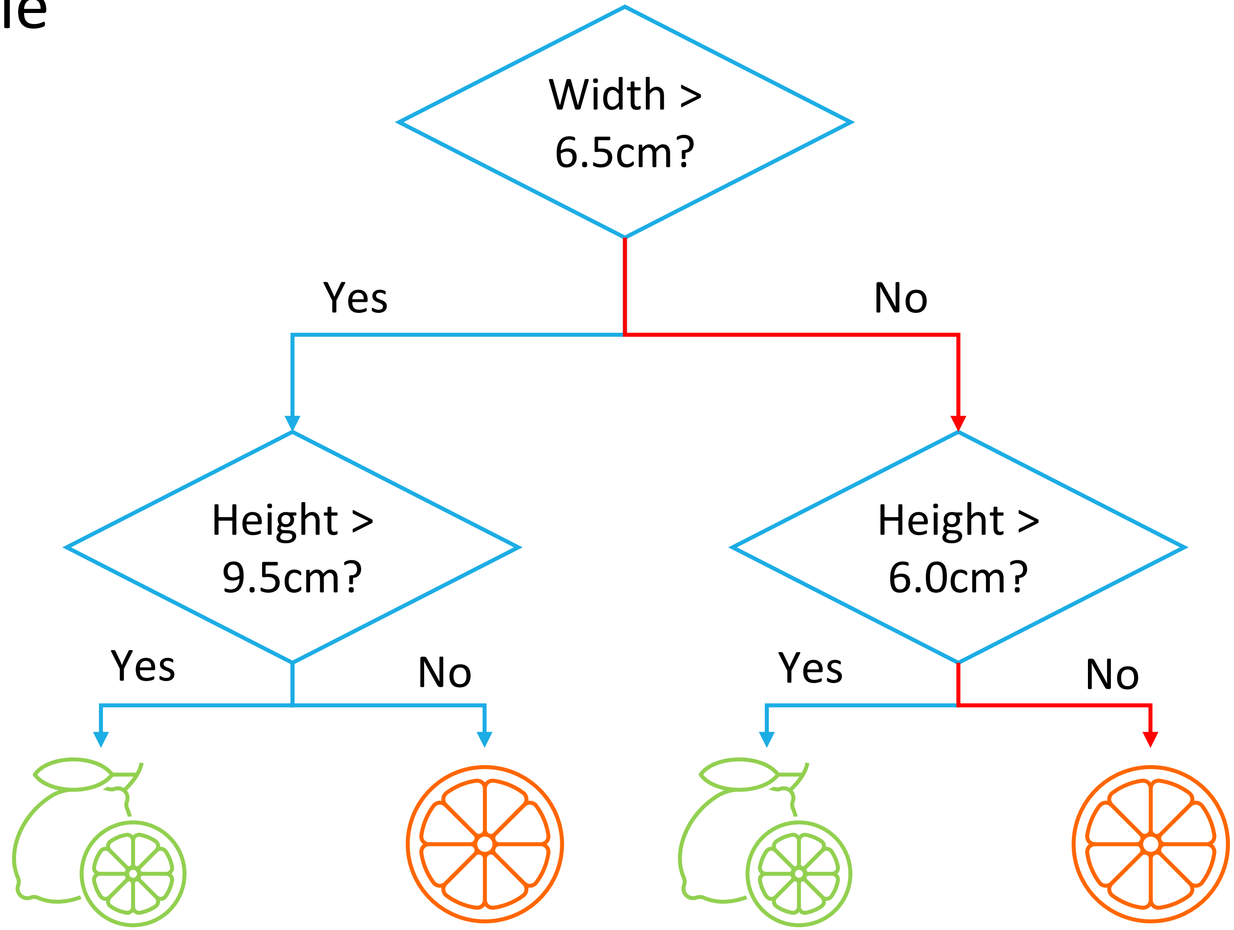
Classification idea

- Test example

5 cm



5 cm



Source: Internet

Binary vs. non-binary conditions

- What if attributes are discrete?

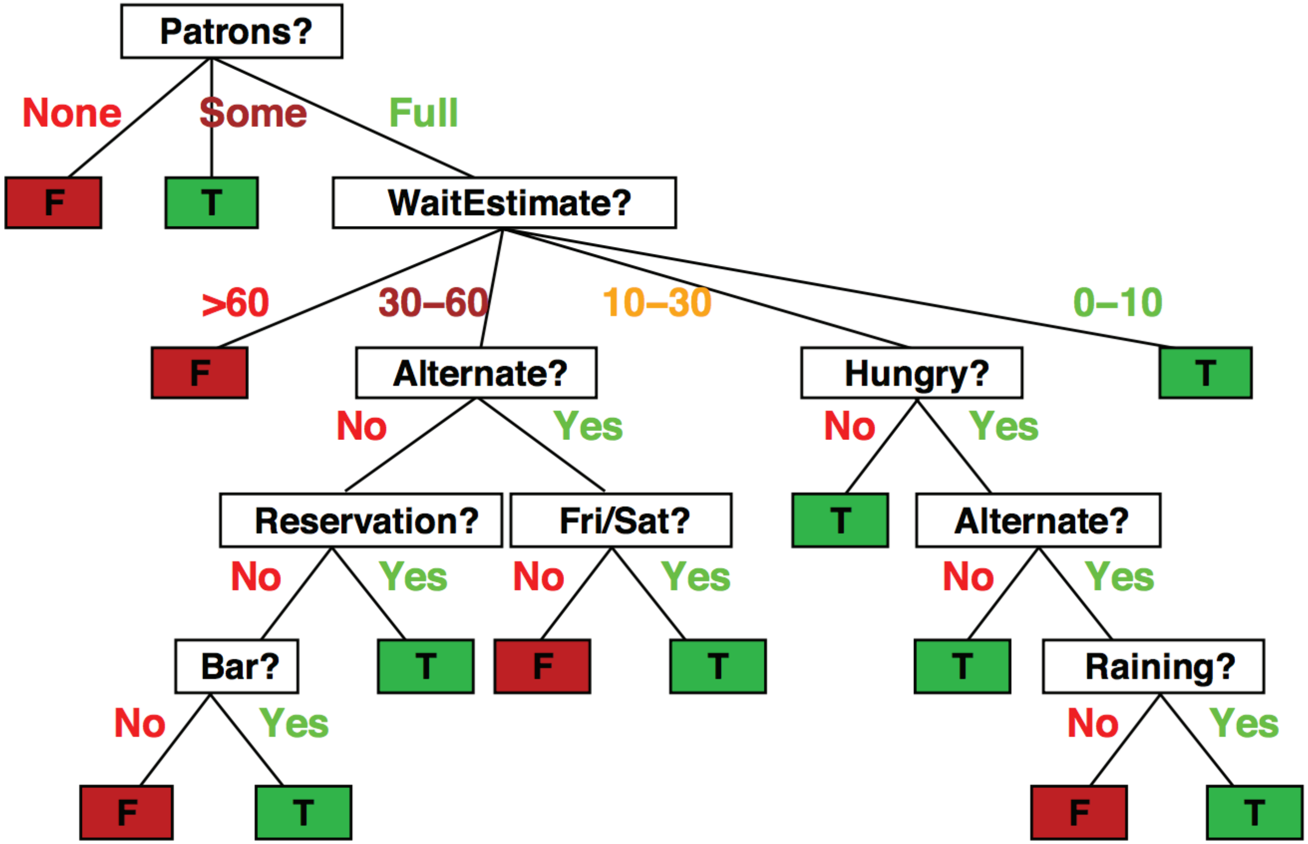
Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*							
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*							
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*							

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 ($, $$, $$$).
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. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Source: Zemel

Binary vs. non-binary conditions

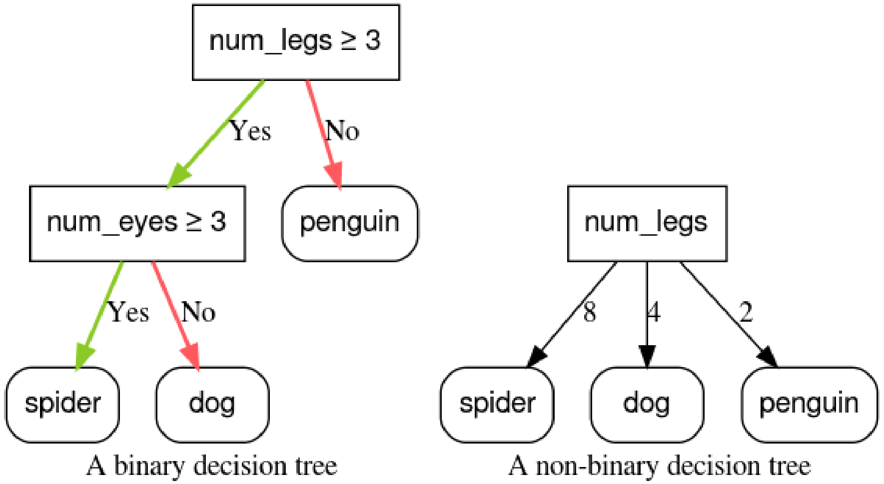
- Decision tree on whether to wait (T) or not (F)



Source: Zemel

Binary vs. non-binary conditions

- Binary conditions: conditions with two possible outcomes, e.g., true or false
- Non-binary conditions have more than two possible outcomes



Binary vs. non-binary decision trees

Source: https://developers.google.com

Outline

- Classification idea
- Learn decision trees ☜
  - Entropy
  - Information gain
  - Decision tree construction algorithm
- Overfitting and pruning

Learn decision tree

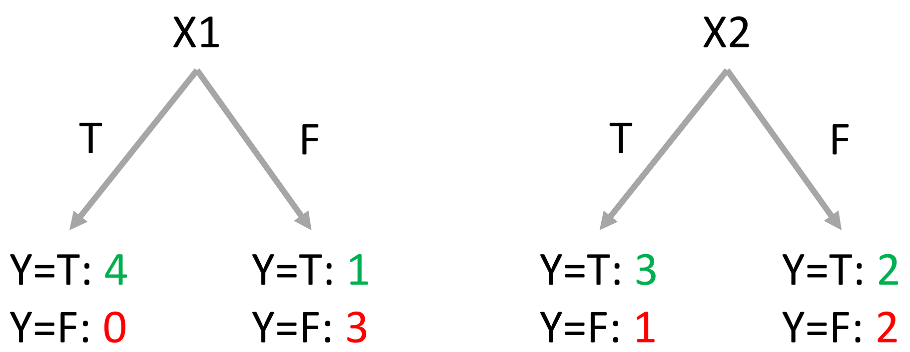
Classification idea

- Pick an attribute, do a simple test
- Conditioned on a choice, pick another attribute, do another test
- In the leaves, assign a class with majority vote
- Do other branches as well

What is the attribute to pick first?

Choose a good attribute

- Which attribute is better to split on, X1 or X2?

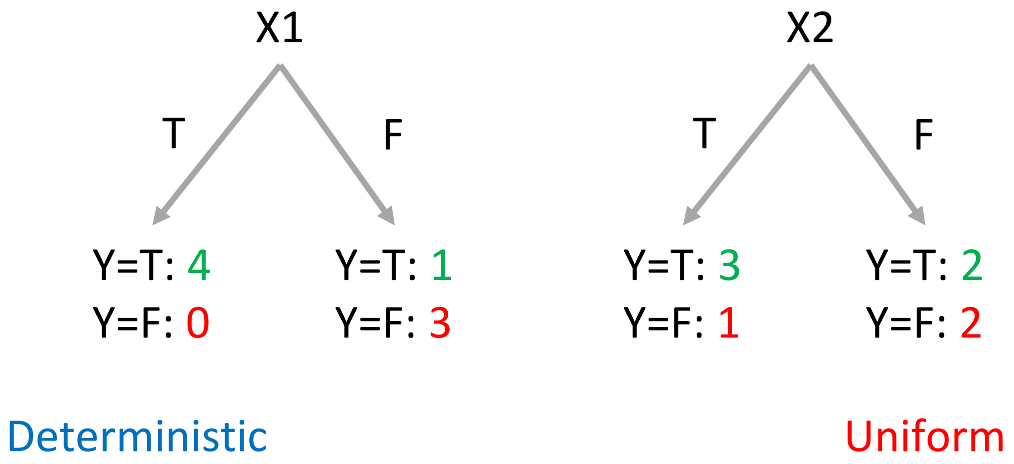


X1	X2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Idea: measure uncertainty by probability distribution of Y at leaves

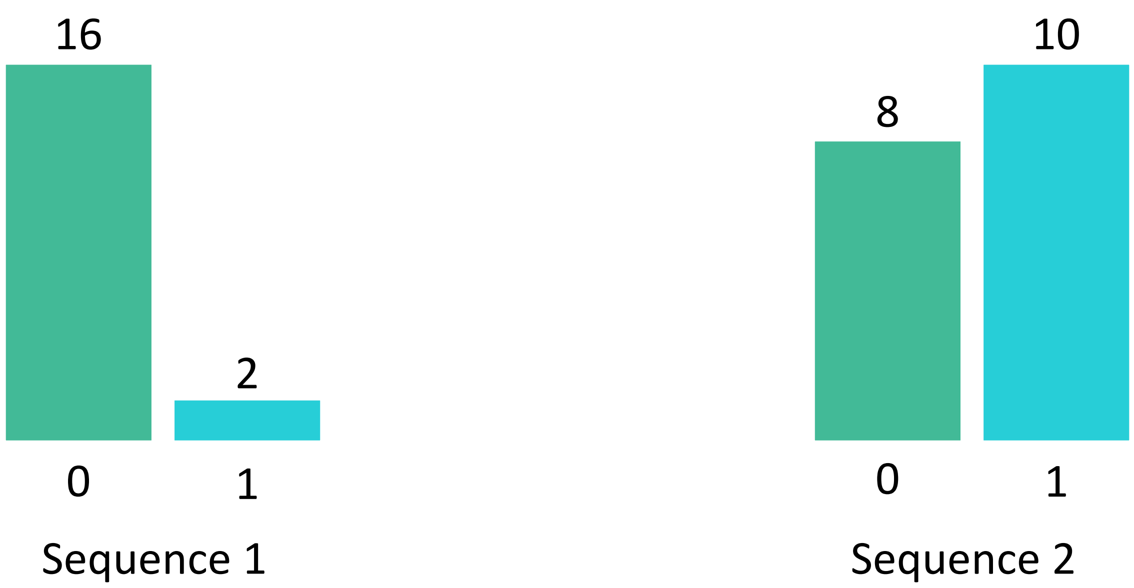
Choose a good attribute

- Which attribute is better to split on, X1 or X2?
  - Deterministic: good (all are true or false; just one class in leaf)
  - Uniform distribution: bad (all classes in leaf equally probable)
  - What about distributions in between?



Choose a good attribute

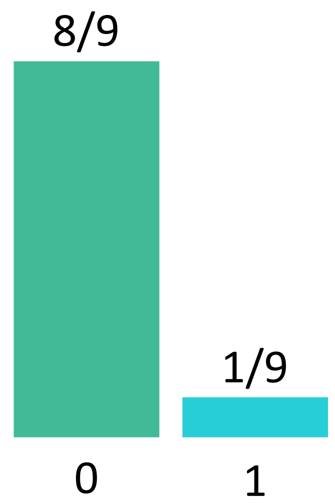
- We flip two different coins
  - Sequence 1: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 … ?
  - Sequence 2: 0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 1 … ?



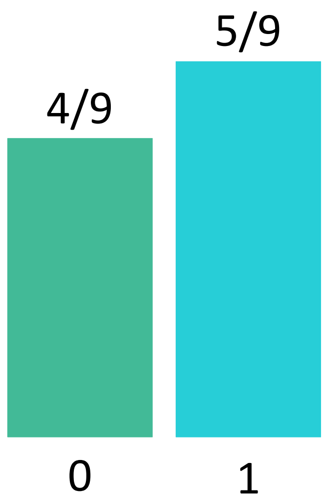
Quantify uncertainty

- Entropy H

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$



$$-\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9} \approx \frac{1}{2}$$

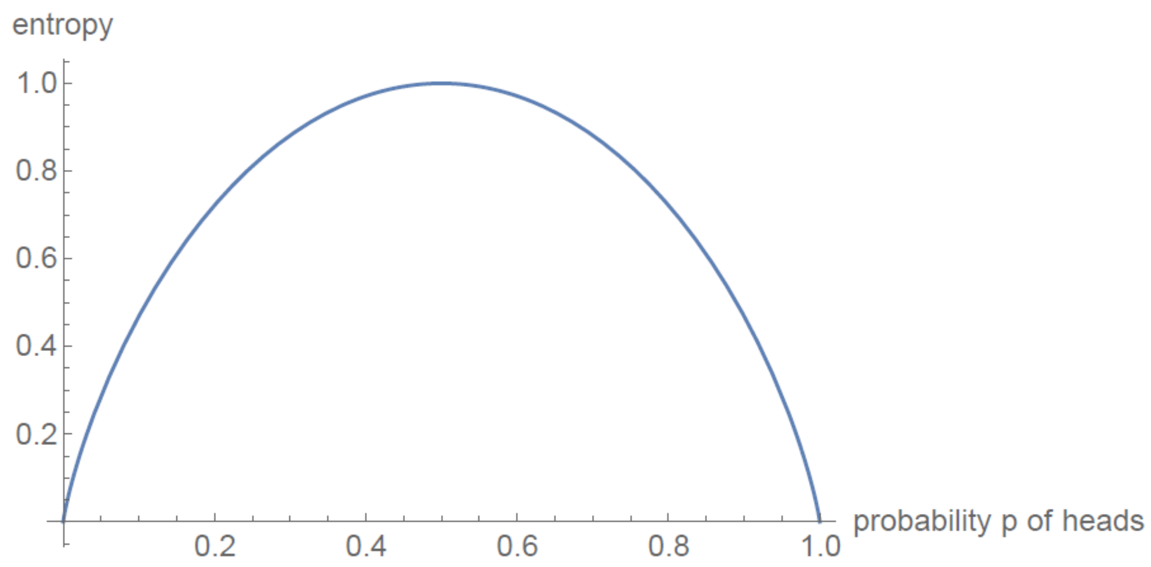


$$-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} \approx 0.99$$

- How easy can we guess a new value in the sequence?
- How much information does it convey?

Quantify uncertainty

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

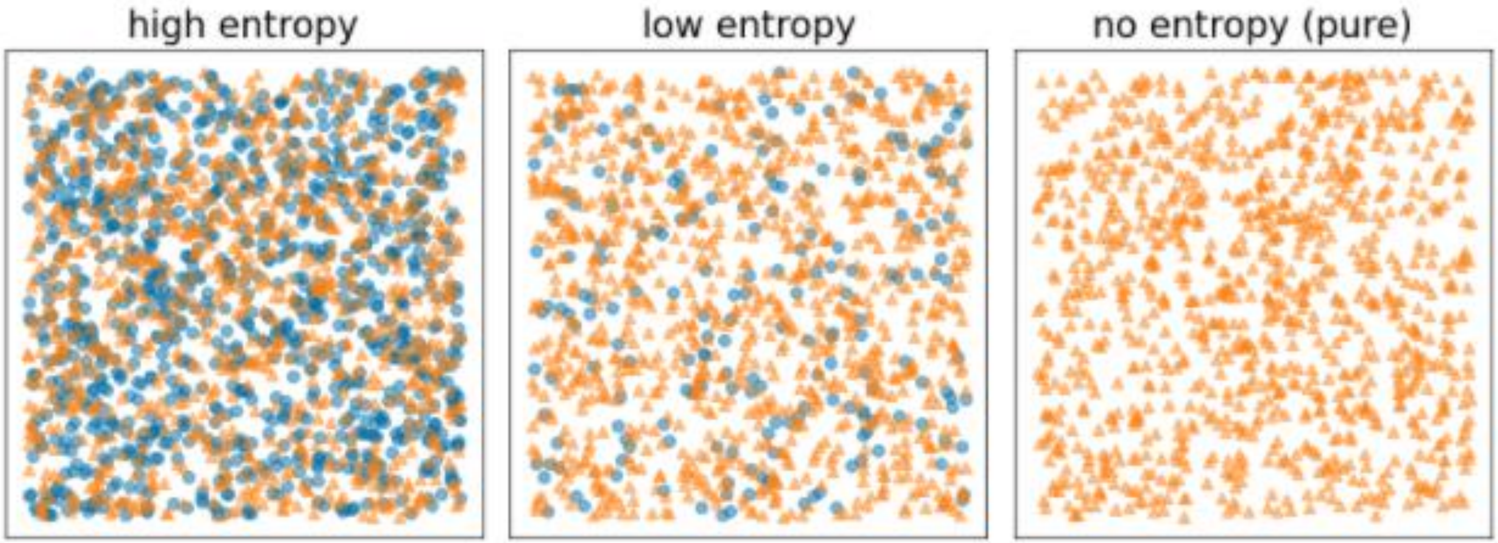


Source: Zemel

Entropy

- Entropy measures how states of disorder, randomness or uncertainty
- High entropy
  - Variable has a uniform like distribution
  - Flat histogram
  - Values sampled from it are less predictable
- Low entropy
  - Distribution of variable has many peaks and valleys
  - Histogram has many lows and highs
  - Values sampled from it are more predictable

Entropy



Three different levels of entropy

Source: https://developers.google.com

Entropy of a Joint Distribution

- Example: X = {Raining, Not raining}, Y = {Cloudy, Not cloudy}

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

$$
\begin{aligned}
H(X,Y) &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y) \\
&= -\frac{24}{100}\log_2\frac{24}{100} - \frac{1}{100}\log_2\frac{1}{100} - \frac{25}{100}\log_2\frac{25}{100} - \frac{50}{100}\log_2\frac{50}{100} \\
&\approx 1.56
\end{aligned}
$$

Specific Conditional Entropy

- Example: X = {Raining, Not raining}, Y = {Cloudy, Not cloudy}

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- What is the entropy of cloudiness Y, given that it is raining?

$$H(Y, X = x) = -\sum_{y \in Y} p(y|x) \log_2 p(y|x)$$

$$= -\frac{24}{25}\log_2\frac{24}{25} - \frac{1}{25}\log_2\frac{1}{25}$$

$$\approx 0.24$$

- We used: $P(y|x) = \frac{p(x,y)}{p(x)}$, and $P(x) = \sum_y p(x,y)$

Conditional Entropy

- Example: X = {Raining, Not raining}, Y = {Cloudy, Not cloudy}

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- The expected conditional entropy

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

$$= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y|x)$$

Conditional Entropy

- Example: X = {Raining, Not raining}, Y = {Cloudy, Not cloudy}

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- What is the entropy of cloudiness, given the knowledge of if it is raining?

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

$$= \frac{1}{4} H(\text{cloudy}|\text{raining}) + \frac{3}{4} H(\text{cloudy}|\text{not raining})$$

$$\approx 0.75$$

Information Gain

- Example: X = {Raining, Not raining}, Y = {Cloudy, Not cloudy}

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- How much information about cloudiness do we get by discovering if it is raining?

$$
\begin{aligned}
IG(Y|X) &= H(Y) - H(Y|X) \\
&\approx 0.25
\end{aligned}
$$

Information Gain

- Information gain in Y due to X: IG(Y | X)
- If X is completely uninformative about Y : IG (Y | X) = 0
- If X is completely informative about Y : IG (Y | X) = H(Y)
- How can we use this to construct our decision tree?

Construct decision tree

- Make use of information gain to partition data samples
- At each level, we need to choose
  - Which variable to split
  - Possibly where to split it
- Choose them based on how much information we would gain from the decision
  - Choose attribute that gives the highest gain

Decision tree construction algorithm

- Step 1: Pick an attribute to split at a non-terminal node
- Step 2: Split examples into groups based on attribute value
- Step 3: For each group
  - If no example, return majority from parent
  - Else if all examples in same class, return class
  - Else go to Step 1

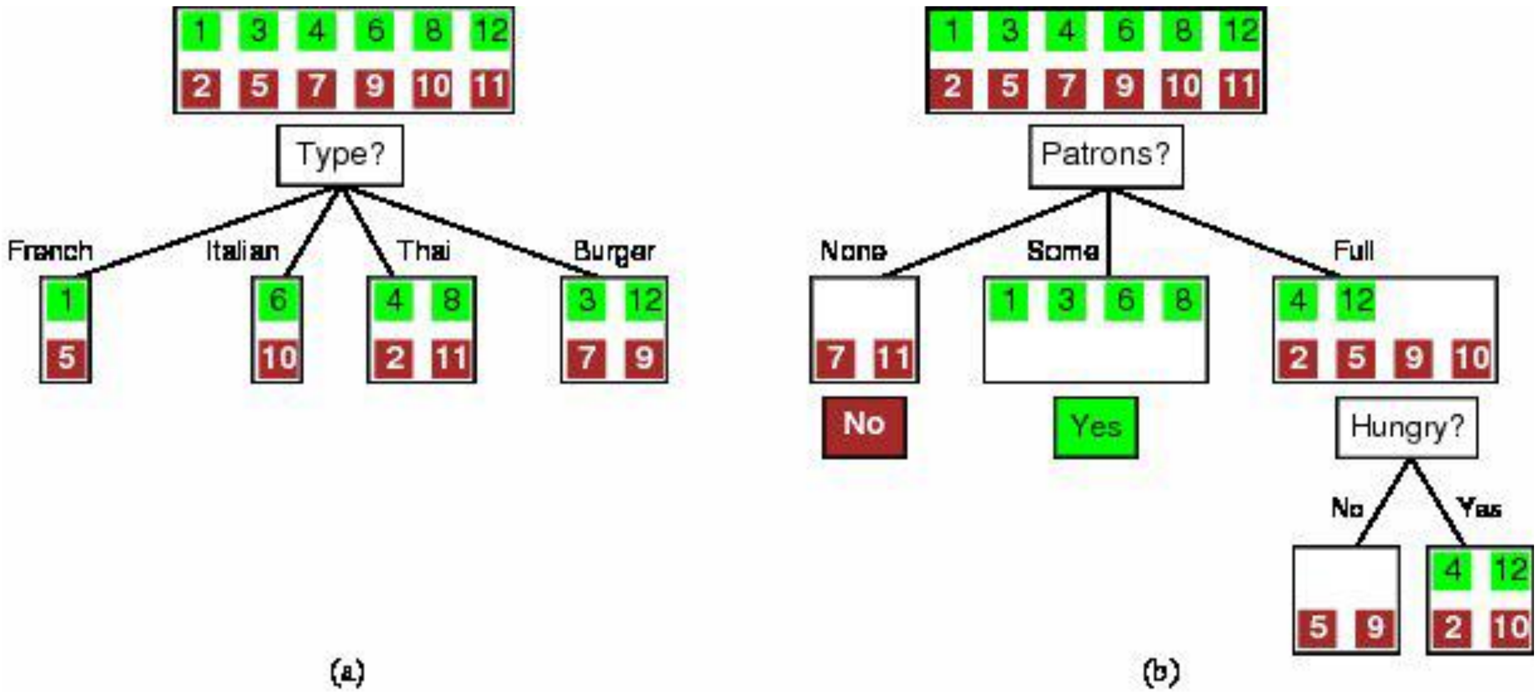
Example

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	$y_1 =$ Yes
$\mathbf{x}_2$	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	$y_2 =$ No
$\mathbf{x}_3$	No	Yes	No	No	Some	$	No	No	Burger	0–10	$y_3 =$ Yes
$\mathbf{x}_4$	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10–30	$y_4 =$ Yes
$\mathbf{x}_5$	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 =$ No
$\mathbf{x}_6$	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	$y_6 =$ Yes
$\mathbf{x}_7$	No	Yes	No	No	None	$	Yes	No	Burger	0–10	$y_7 =$ No
$\mathbf{x}_8$	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	$y_8 =$ Yes
$\mathbf{x}_9$	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 =$ No
$\mathbf{x}_{10}$	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	$y_{10} =$ No
$\mathbf{x}_{11}$	No	No	No	No	None	$	No	No	Thai	0–10	$y_{11} =$ No
$\mathbf{x}_{12}$	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	$y_{12} =$ Yes

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 ($, $$, $$$).
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.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Source: Zemel

Example



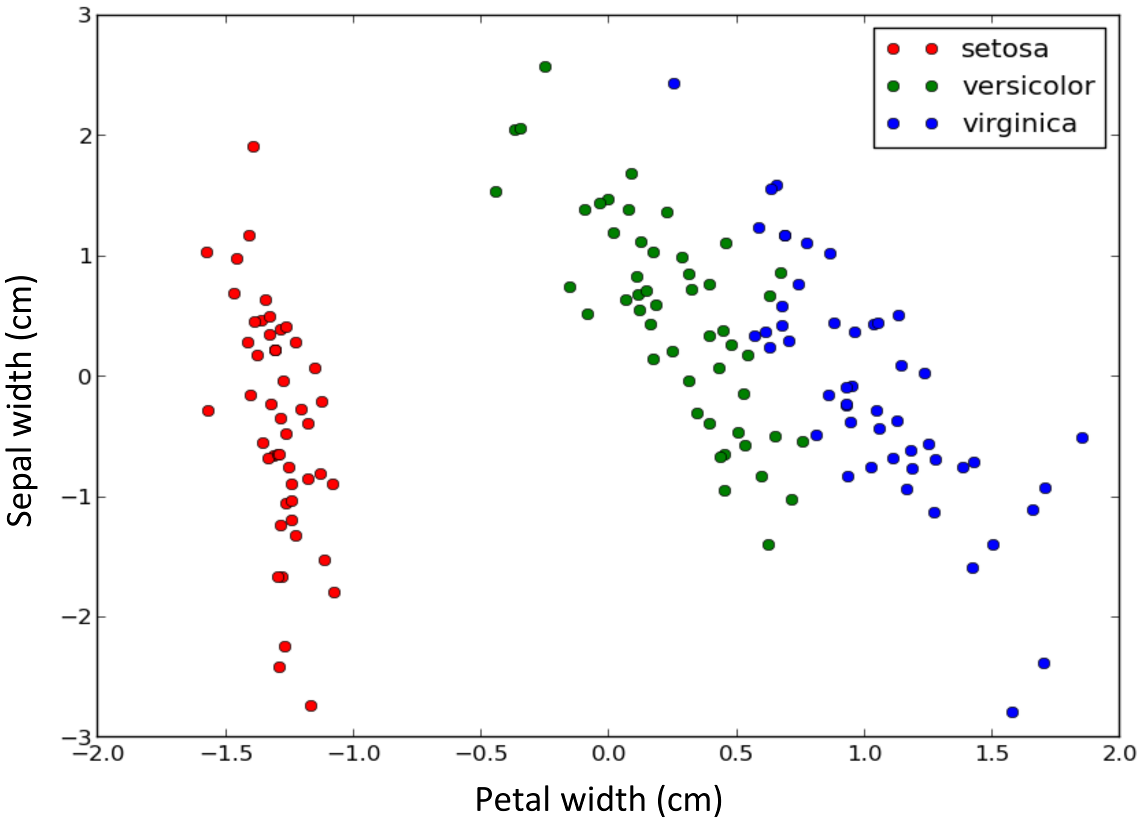
$$IG(Y) = H(Y) - H(Y|X)$$

$$IG(type) = 1 - \left[\frac{2}{12}H(Y|\text{Fr.}) + \frac{2}{12}H(Y|\text{It.}) + \frac{4}{12}H(Y|\text{Thai}) + \frac{4}{12}H(Y|\text{Bur.})\right] = 0$$

$$IG(Patrons) = 1 - \left[\frac{2}{12}H(0,1) + \frac{4}{12}H(1,0) + \frac{6}{12}H(\frac{2}{6},\frac{4}{6})\right] \approx 0.541$$

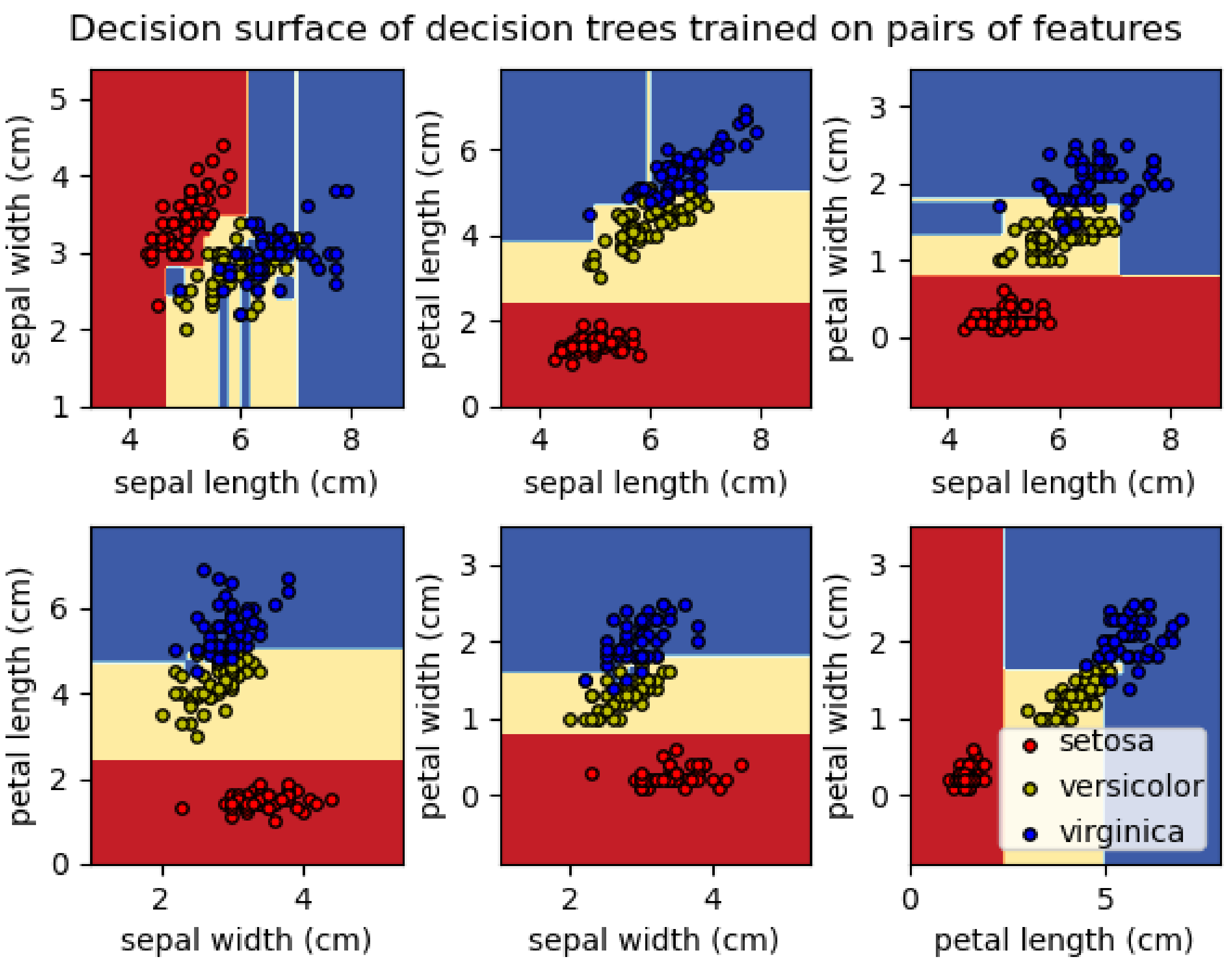
Source: Zemel

Example on Iris dataset



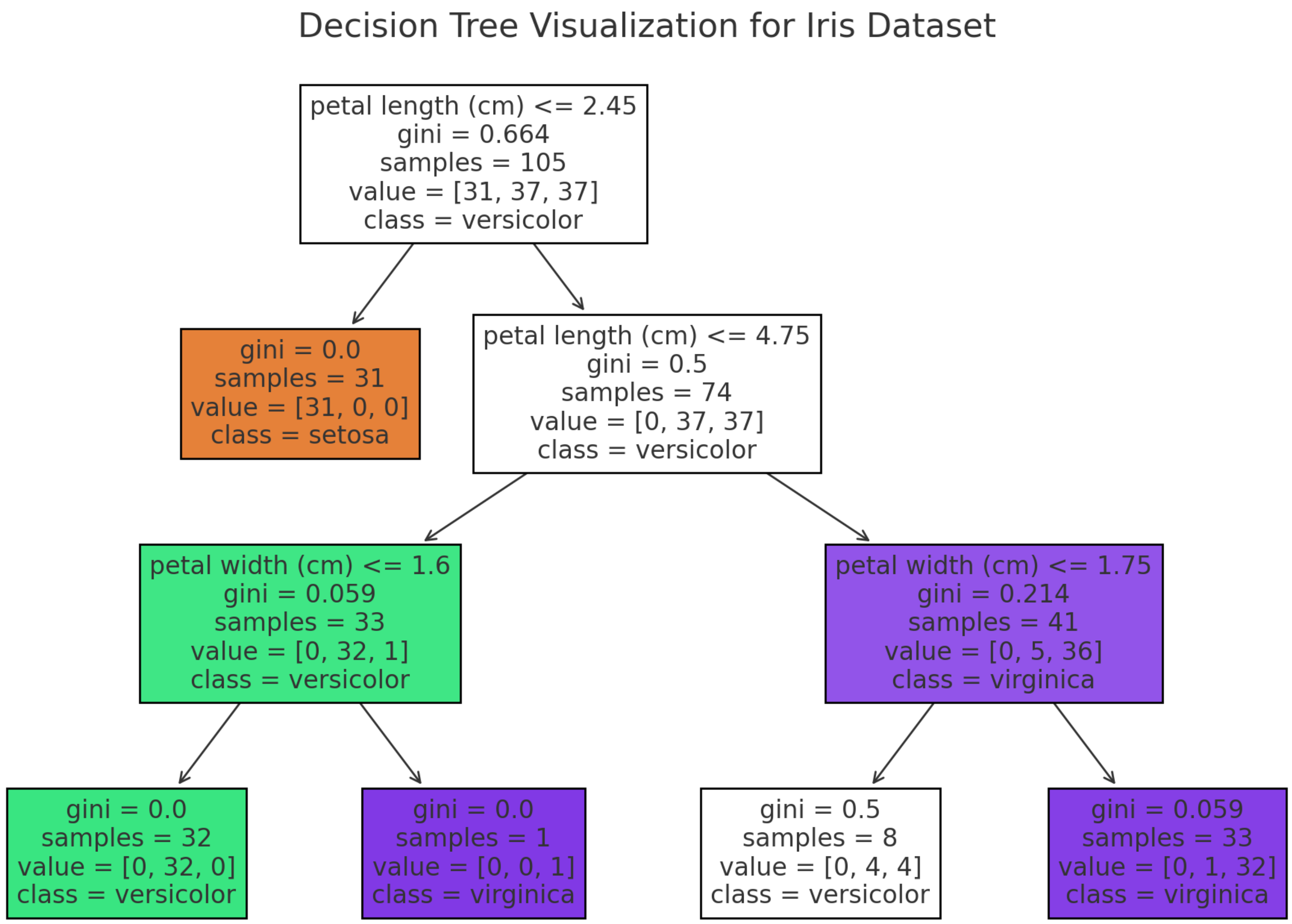
Source: Internet

Example on Iris dataset



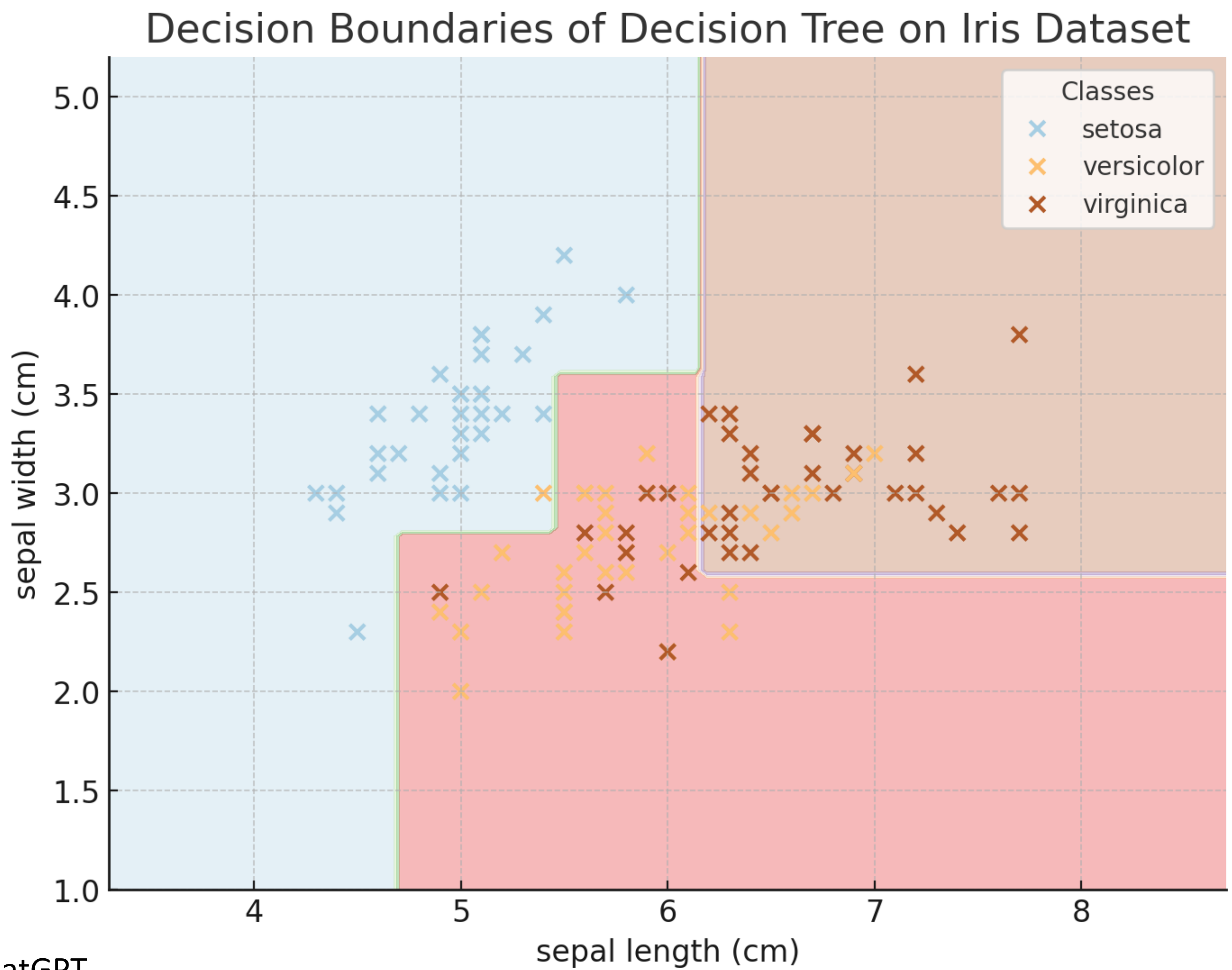
Source: https://scikit-learn.org

Example on Iris dataset



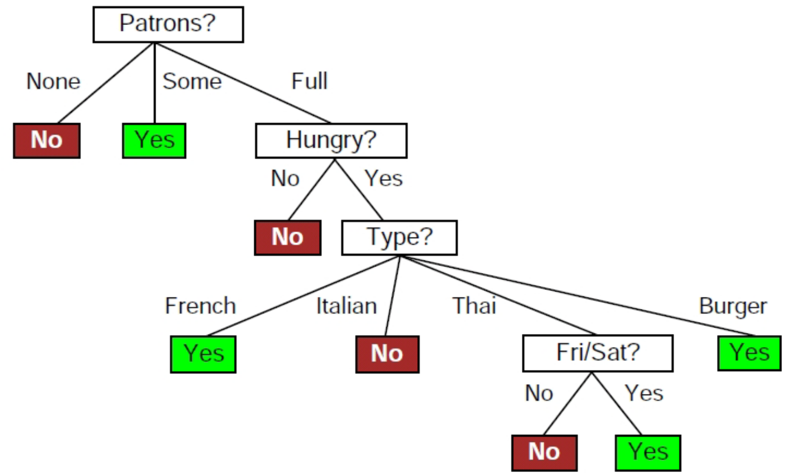
Source: ChatGPT

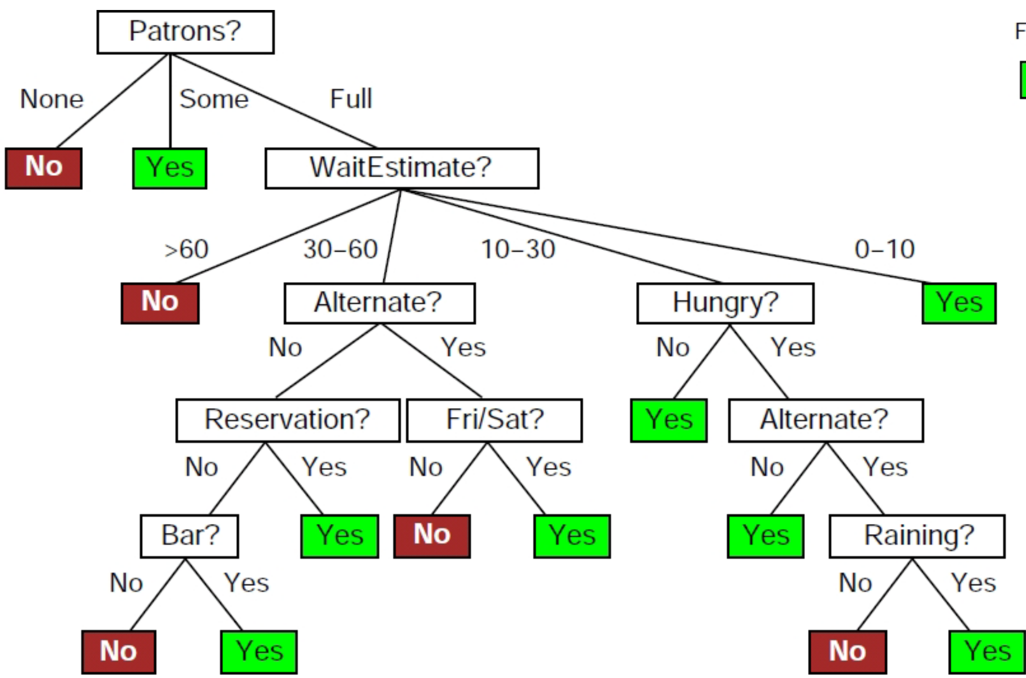
Example on Iris dataset



Source: ChatGPT

Which tree is better?





What makes a good tree

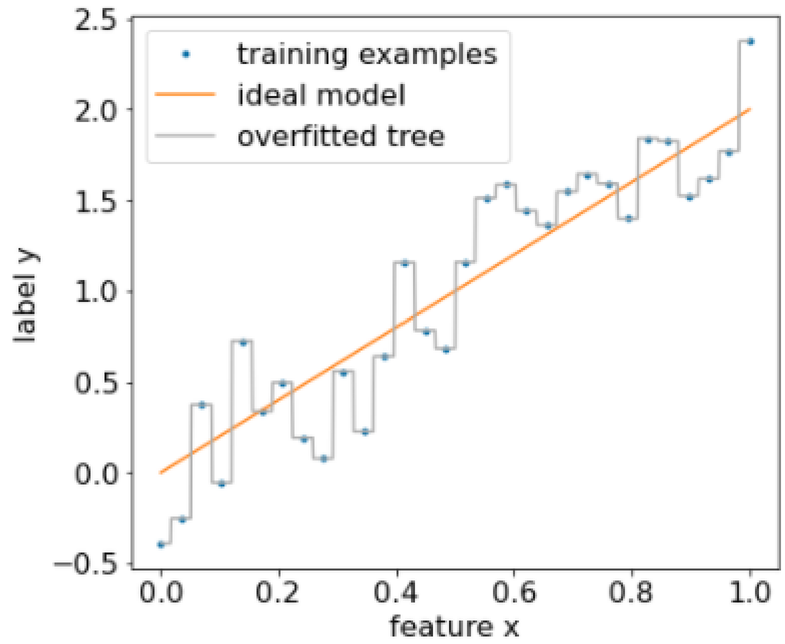
- Not too small: need to handle important but possibly subtle distinctions in data
- Not too big:
  - Computational efficiency
  - Avoid overfitting training set
- Find the simplest hypothesis, i.e., smallest tree that fits the observations
- Small trees with informative nodes near the root

Outline

- Classification idea
- Learn decision trees
  - Entropy
  - Information gain
  - Decision tree construction algorithm
- Overfitting and pruning ☜

Overfitting and pruning

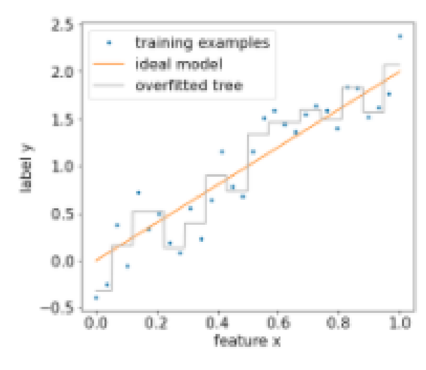
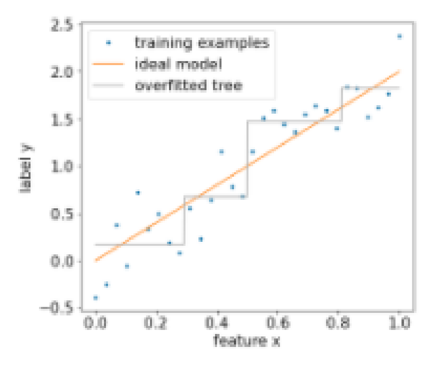
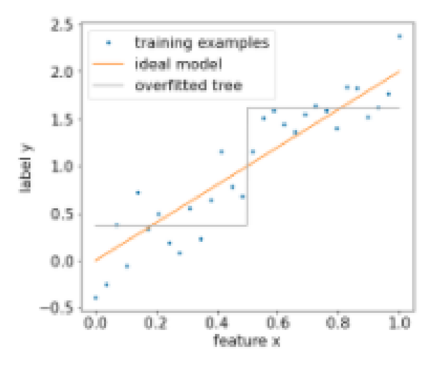
- If the dataset contains noise, this tree will overfit to the data and show poor test accuracy



Source: https://developers.google.com

Overfitting and pruning

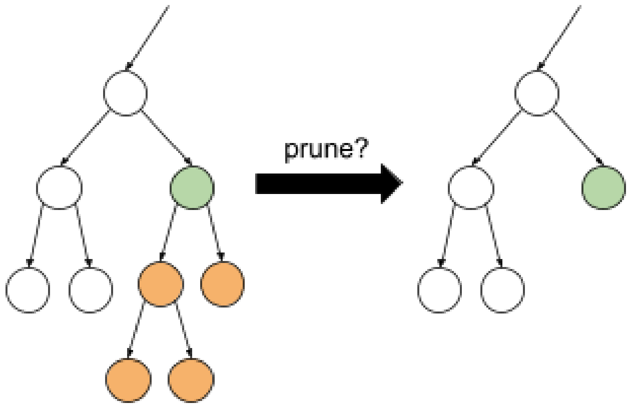
- To limit overfitting a decision tree, apply one or both of the following regularization criteria:
  - Set a maximum depth: Prevent decision trees from growing past a maximum depth, such as 10.
  - Set a minimum number of examples in leaf: A leaf with less than a certain number of examples will not be considered for splitting.

minimum number of examples in leaf: 2	minimum number of examples in leaf: 5	minimum number of examples in leaf: 10
		

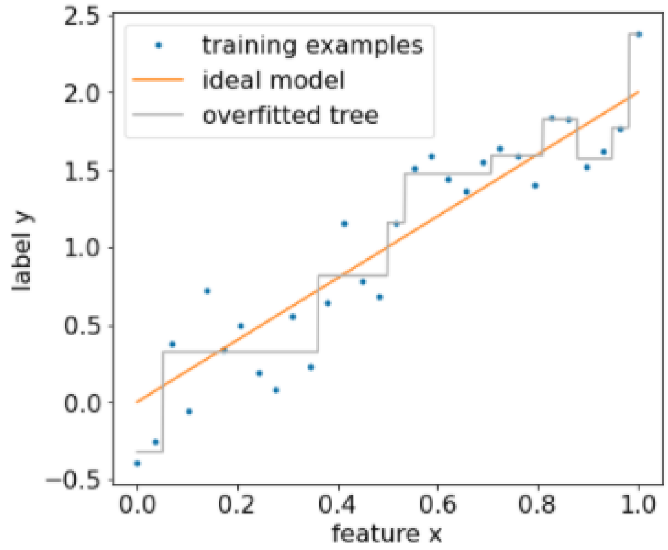
Source: https://developers.google.com

Overfitting and pruning

- Pruning: selectively remove certain branches, that is, by converting certain non-leaf nodes to leaves.
  - Common solution: use a validation dataset to select branches to remove.
  - That is, if removing a branch improves the quality of the model on the validation dataset, then the branch is removed.



Source: https://developers.google.com



Effect of using 20% dataset as validation for pruning

References

This slide borrowed ideas from

- CSC 411: Lecture 06: Decision Trees, Richard Zemel, Raquel Urtasun and Sanja Fidler, University of Toronto

References

- https://developers.google.com/machine-learning/decision-forests