Decision Trees

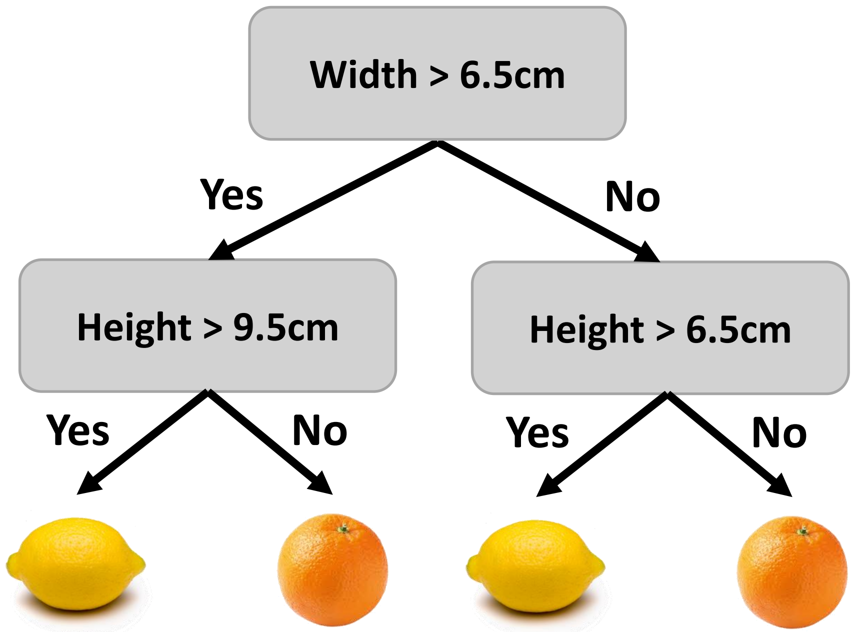
Data Intelligence and Learning (DIAL) Lab
Prof. Jongwuk Lee

Decision Tree Basics

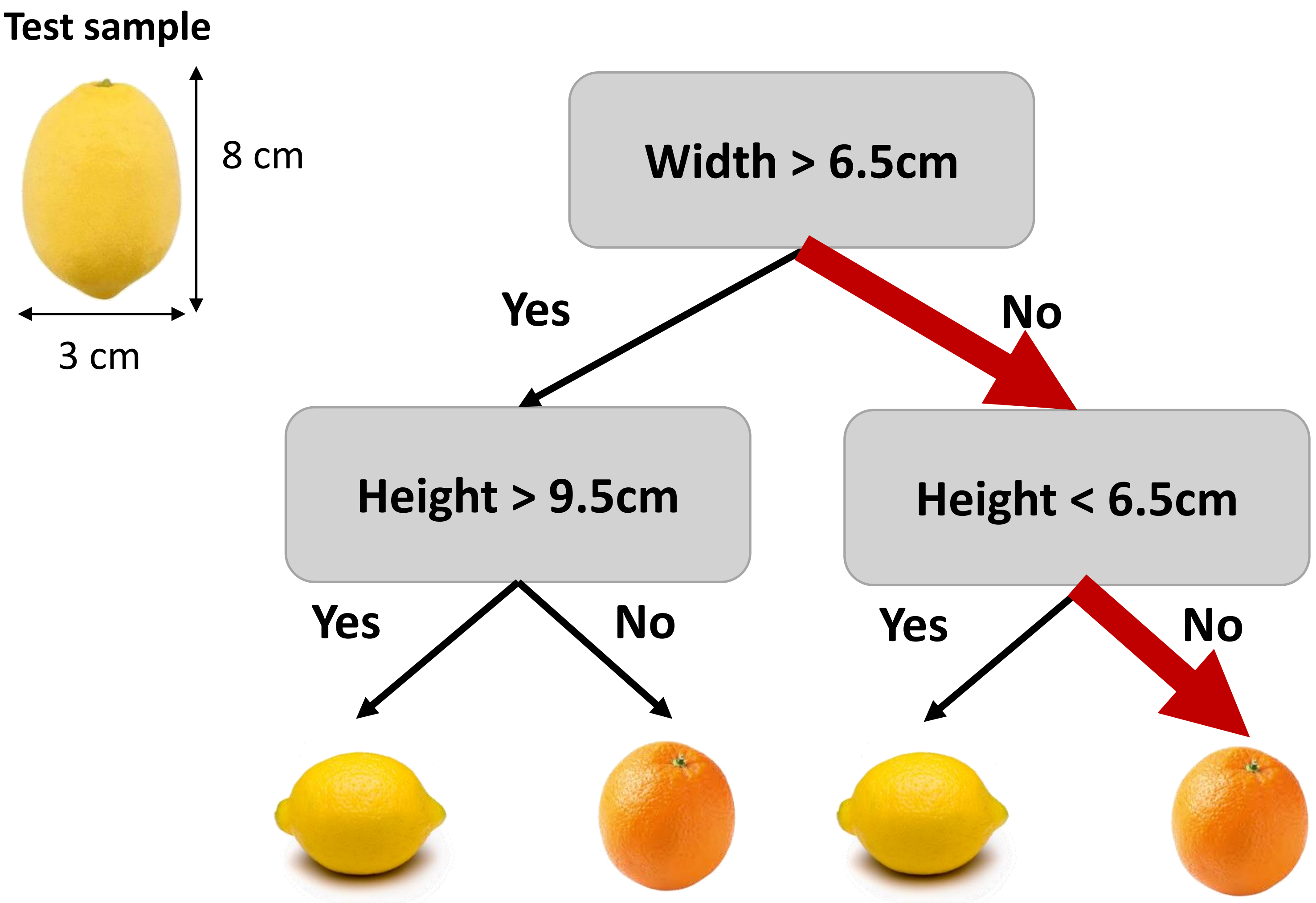
What are Decision Trees?

➢ **Decision trees make predictions by recursively splitting into different attributes.**

- Each **internal node** represents an **attribute** at each stage.
- Each **branch** represents an **attribute value**.
- Each **leaf node** represents a **class** label.
- The path from the root to a leaf node represents a **classification rule**.

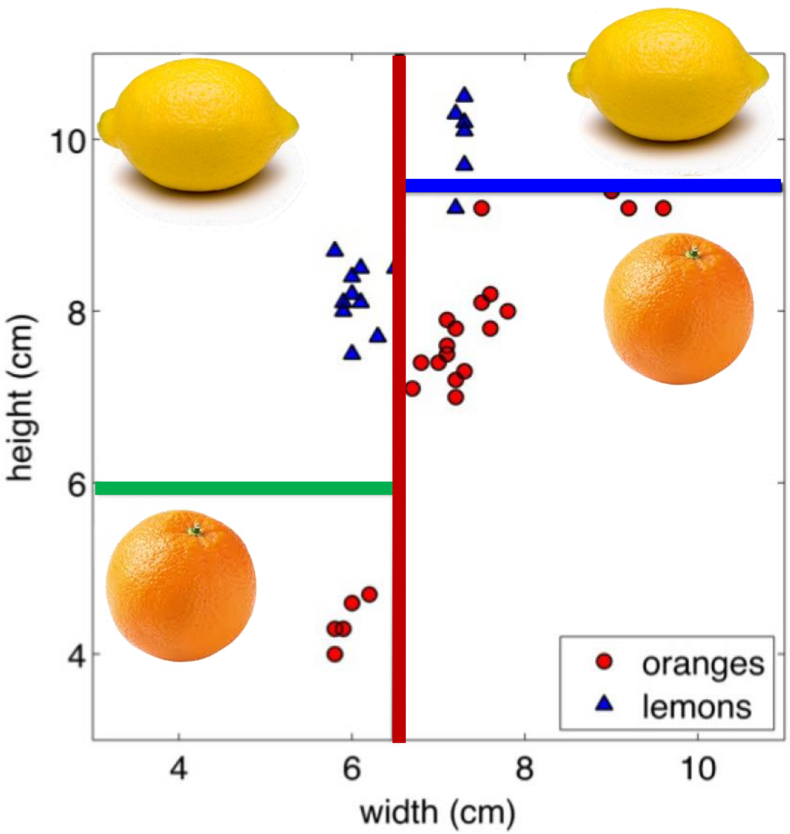


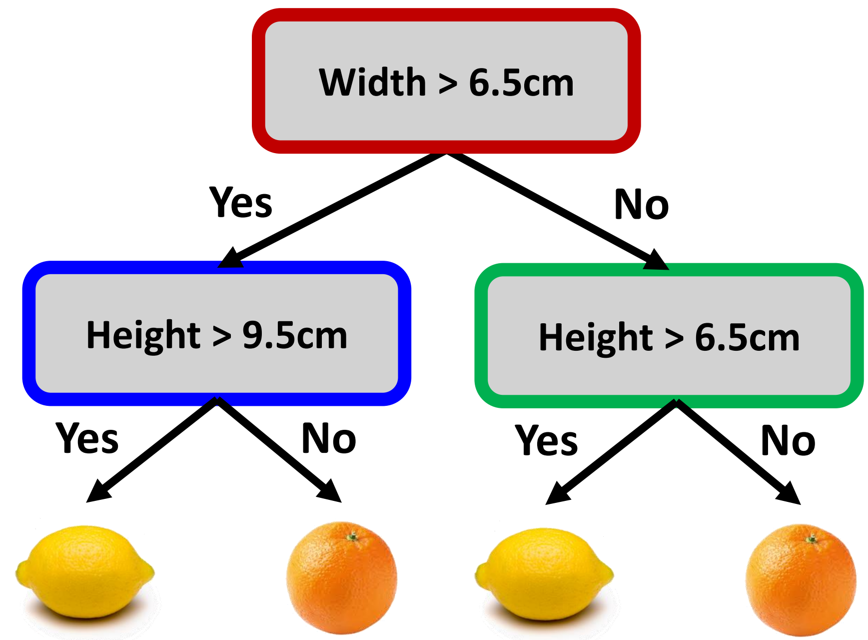
What are Decision Trees?



What are Decision Trees?

- **For a continuous attribute, we split it with a specific value.**
  - The input space is **recursively** divided into **two regions** parallel to axes.

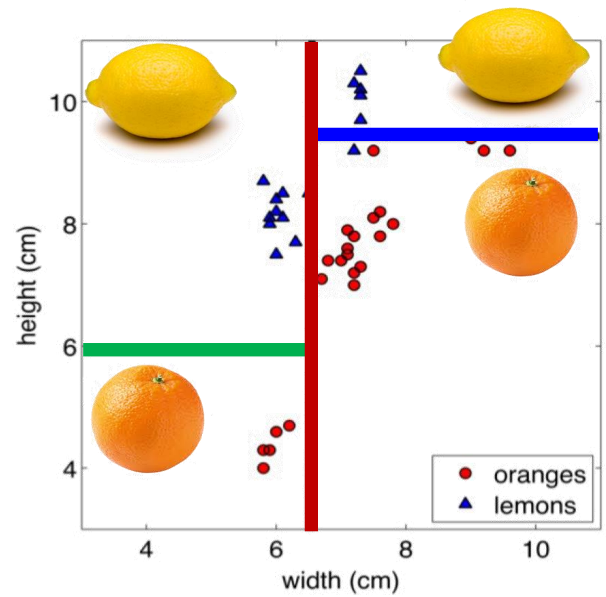




Classification and Regression

- **Each path from the root to a leaf node corresponds to a region R_m of input space.**
  - Let $\{(x^{(m_1)}, y^{(m_1)}), \dots, (x^{(m_k)}, y^{(m_k)})\}$ be training samples in R_m.

- **Classification tree**
  - Discrete output
  - Leaf value represents the **most common value** in $\{y^{(m_1)}, \dots, y^{(m_k)}\}$.



- **Regression tree**
  - Continuous output
  - Leaf value represents the **mean value** in $\{y^{(m_1)}, \dots, y^{(m_k)}\}$.

Classification: Good vs. Bad

- We want to construct a classification tree to classify heroes as good or bad according to their appearance.

Training data

	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batman	Male	Yes	Yes	No	Yes	No	**Good**
Robin	Male	Yes	Yes	No	No	No	**Good**
Alfred	Male	No	No	Yes	No	No	**Good**
Penguin	Male	No	No	Yes	No	Yes	**Bad**
Catwoman	Female	Yes	No	No	Yes	No	**Bad**
Joker	Male	No	No	No	No	No	**Bad**

Classification: Good vs. Bad

- A hero is a male who has a mask, a cape, and no tie.
- Q: Is he a good or bad man?

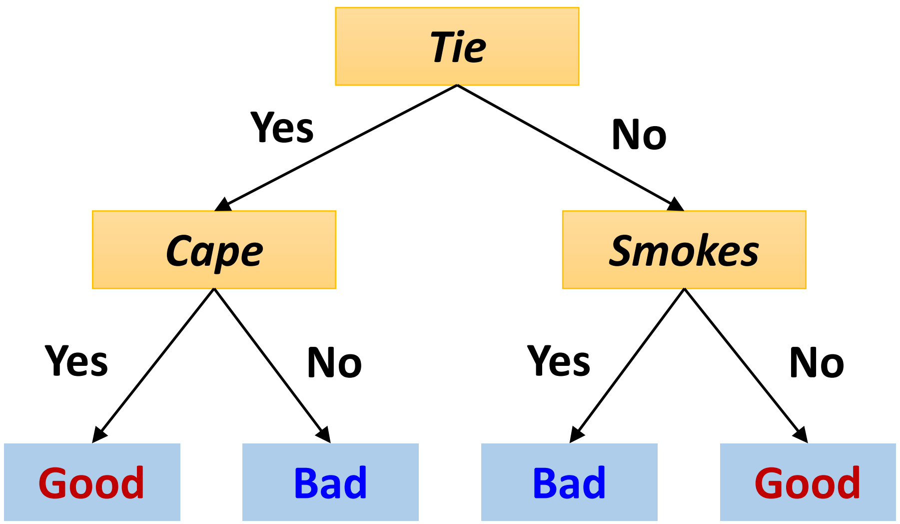
- How do we predict whether he is good or bad?

Training data

	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batman	Male	Yes	Yes	No	Yes	No	**Good**
Robin	Male	Yes	Yes	No	No	No	**Good**
Alfred	Male	No	No	Yes	No	No	**Good**
Penguin	Male	No	No	Yes	No	Yes	**Bad**
Catwoman	Female	Yes	No	No	Yes	No	**Bad**
Joker	Male	No	No	No	No	No	**Bad**

Classification: Good vs. Bad

- Predict a new hero (unlabeled data) as good or bad.



Test data

	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batgirl	Female	Yes	Yes	Yes	Yes	No	??
Riddler	Male	Yes	No	No	No	No	??

How to Learn the Decision Tree?

- **Finding an optimal decision tree that correctly classifies a training set is an NP-complete problem.**
  - Note: the optimal decision tree means the **smallest decision tree**.

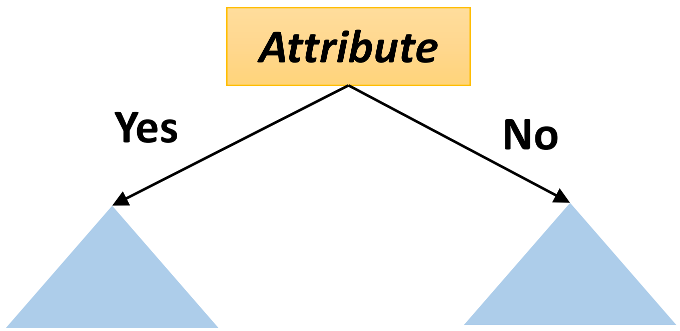
- **Use a greedy heuristic!**
  - Split on the best attribute at each stage.

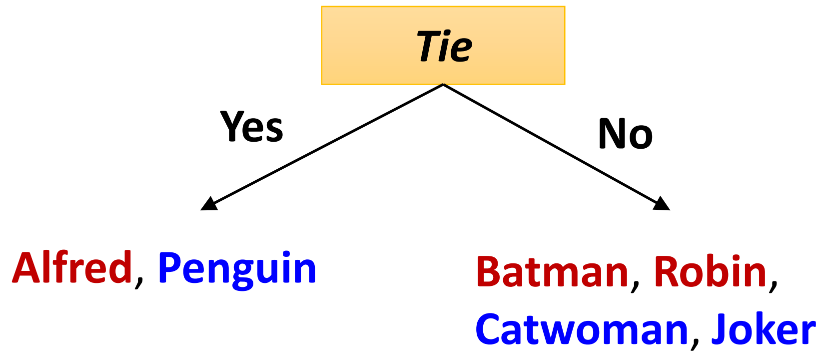
- **Which attribute is the best?**

- **When should we stop?**

Choosing a Good Split

- Choosing a decision rule to split data into disjoint subsets

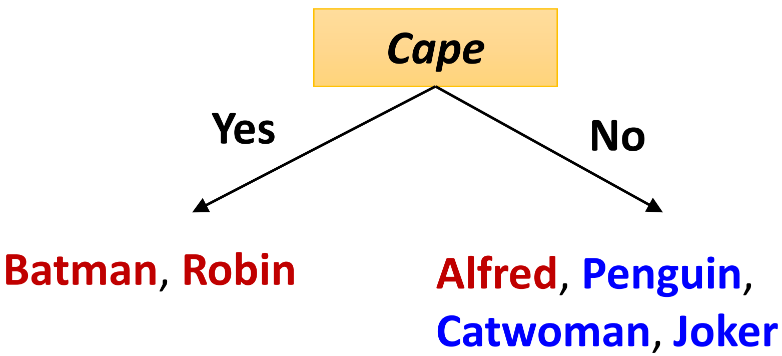


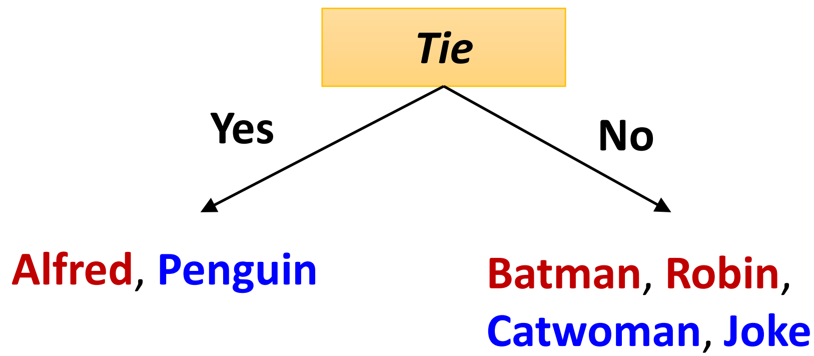


	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batman	Male	Yes	Yes	No	Yes	No	**Good**
Robin	Male	Yes	Yes	No	No	No	**Good**
Alfred	male	No	No	Yes	No	No	**Good**
Penguin	Male	No	No	Yes	No	Yes	**Bad**
Catwoman	Female	Yes	No	No	Yes	No	**Bad**
Joker	Male	No	No	No	No	No	**Bad**

Choosing a Good Split

- Which attribute is better?

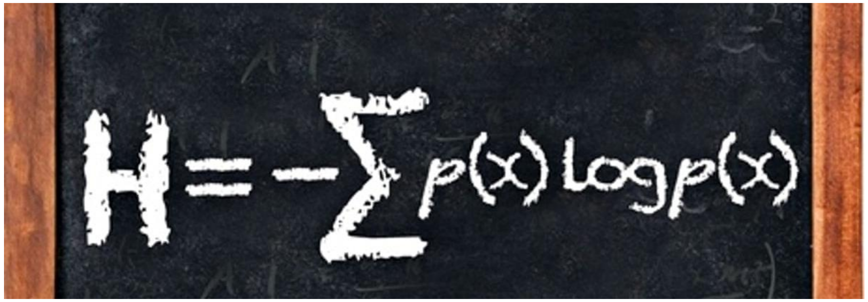


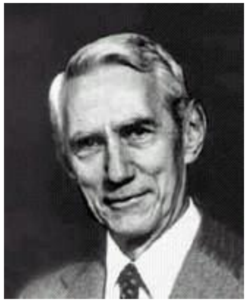


Choosing a Good Split

- **How do we measure uncertainty in prediction for a leaf node?**
  - **All samples** in the leaf node have the same class: **good, low uncertainty**
  - **Each class has the same samples** in the leaf node: **bad, high uncertainty**

- **Idea: Define the probability distribution and use information theory to measure uncertainty.**



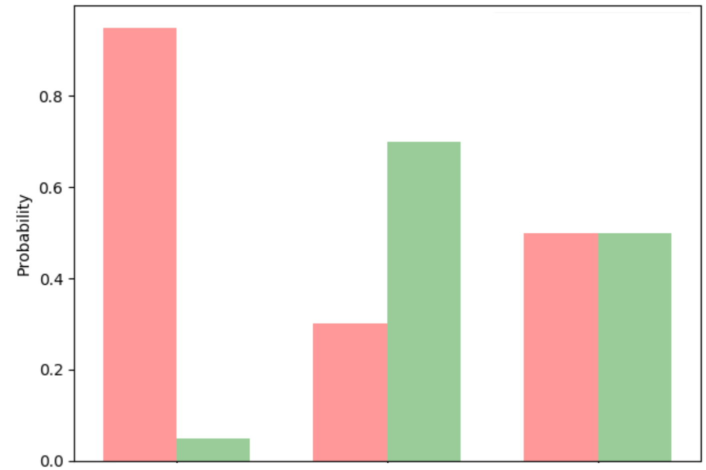


Information Entropy

- Given a discrete random variable X, information entropy is defined as follows.

$$H(X) = \mathbb{E}_{x\sim P(x)}[-\log_2 p(x)] = -\sum_{x\in X} p(x)\log_2 p(x)$$

- Two extreme cases
  - Samples only have one class.
  - Samples are divided into each class.



A *Low entropy* **B** **C** *High entropy*

Information Entropy

- **Measuring the degree of uncertainty**

$$H(X) = -P_1 \log_2 P_1 - P_0 \log_2 P_0$$

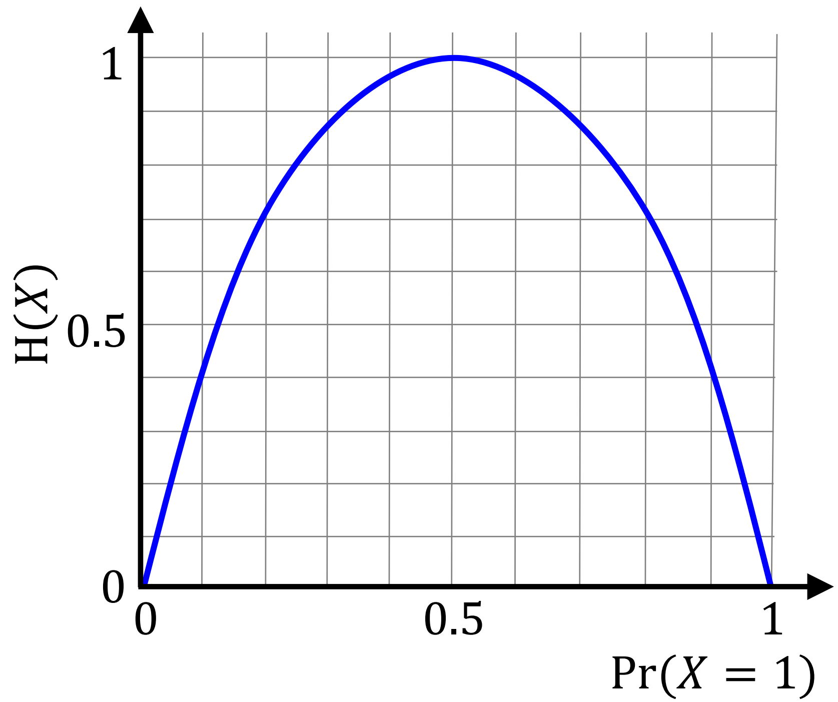
P_1 = The probability that 1 appears in the set
P_0 = The probability that 0 appears in the set

$A = \{1,1,0,0,0,0,0,0\}$

$H(X) = -2/8 \log_2 2/8 - 6/8 \log_2 6/8$

$B = \{1,1,1,1,0,0,0,0\}$

$H(X) = -4/8 \log_2 4/8 - 4/8 \log_2 4/8$



Information Entropy

- Assume that there are two classes $\boldsymbol{P}$ and $\boldsymbol{N}$.

- A dataset $\mathcal{D}$ contains p elements of class $\boldsymbol{P}$ and n elements of class $\boldsymbol{N}$, respectively.

- For the dataset $\mathcal{D}$, the information entropy is computed by:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Conditional Entropy

- **Measuring the average of the entropy for each partition**

$$H(Y \mid X) = \mathbb{E}_{x\sim P(x)}[H(Y \mid X = x)] = \sum_{x\in X} p(x)\, H(Y \mid X = x)$$

- **Which partition is better?**

$$\{1,1,1,1,0,0,0,0\}$$

$$A = \{\{1,1,0,0,0\},\{1,1,0\}\} \qquad \text{vs.} \qquad B = \{\{1,1,1,1,0\},\{0,0,0\}\}$$

Conditional Entropy

$$A = \{\{1,1,0,0,0\},\{1,1,0\}\}$$

$$-2/5 \log_2 2/5 - 3/5 \log_2 3/5 = 0.97$$

$$-2/3 \log_2 2/3 - 1/3 \log_2 1/3 = 0.92$$

$$\frac{5}{8} \times 0.97 + \frac{3}{8} \times 0.92 = 0.95$$

$$B = \{\{1,1,1,1,0\},\{0,0,0\}\}$$

$$-4/5 \log_2 4/5 - 1/5 \log_2 1/5 = 0.72$$

$$-3/3 \log_2 3/3 - 0/3 \log_2 0/3 = 0.00$$

$$\frac{5}{8} \times 0.72 + \frac{3}{8} \times 0.0 = 0.45$$

Information Gain

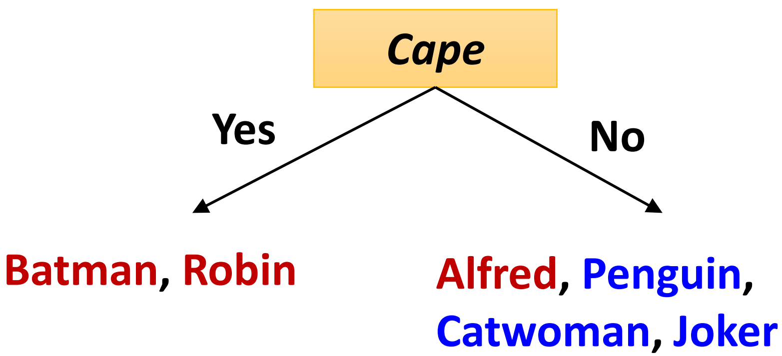
- **The information gain in Y due to A, or the mutual information of Y and A is defined as follows.**

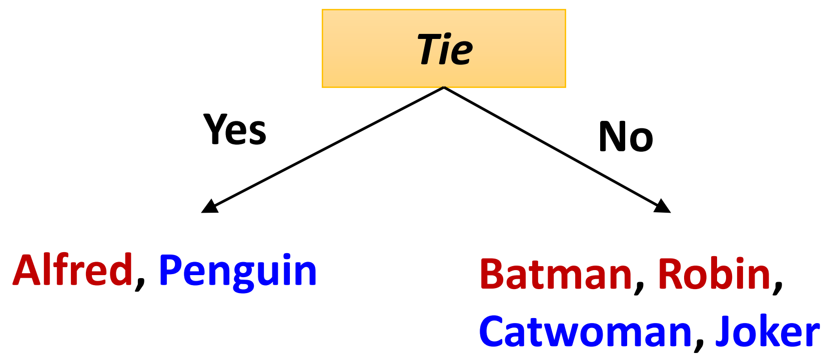
$$IG(Y \mid A) = H(Y) - H(Y \mid A)$$

- **If A is completely informative about Y:** $IG(Y \mid A) = H(Y)$
  - The conditional entropy $H(Y \mid A)$ is 0.
- **If A is completely uninformative about Y:** $IG(Y \mid A) = 0$
  - The conditional entropy $H(Y \mid A)$ is equal to $H(Y)$.

Information Gain

- Suppose that an attribute A is chosen.



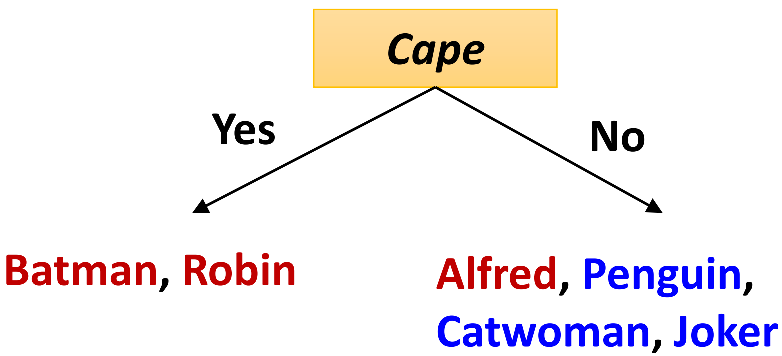


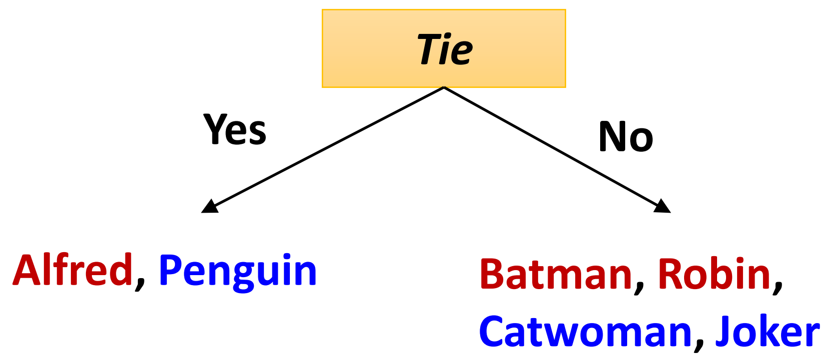
$$H(Y \mid Cape) = \frac{2}{6} \times I(1{,}0) + \frac{4}{6} \times I(3{,}1)$$

$$H(Y \mid Tie) = \frac{2}{6} \times I(1{,}1) + \frac{4}{6} \times I(2{,}2)$$

Information Gain

- **Entropy before branching – entropy after branching**





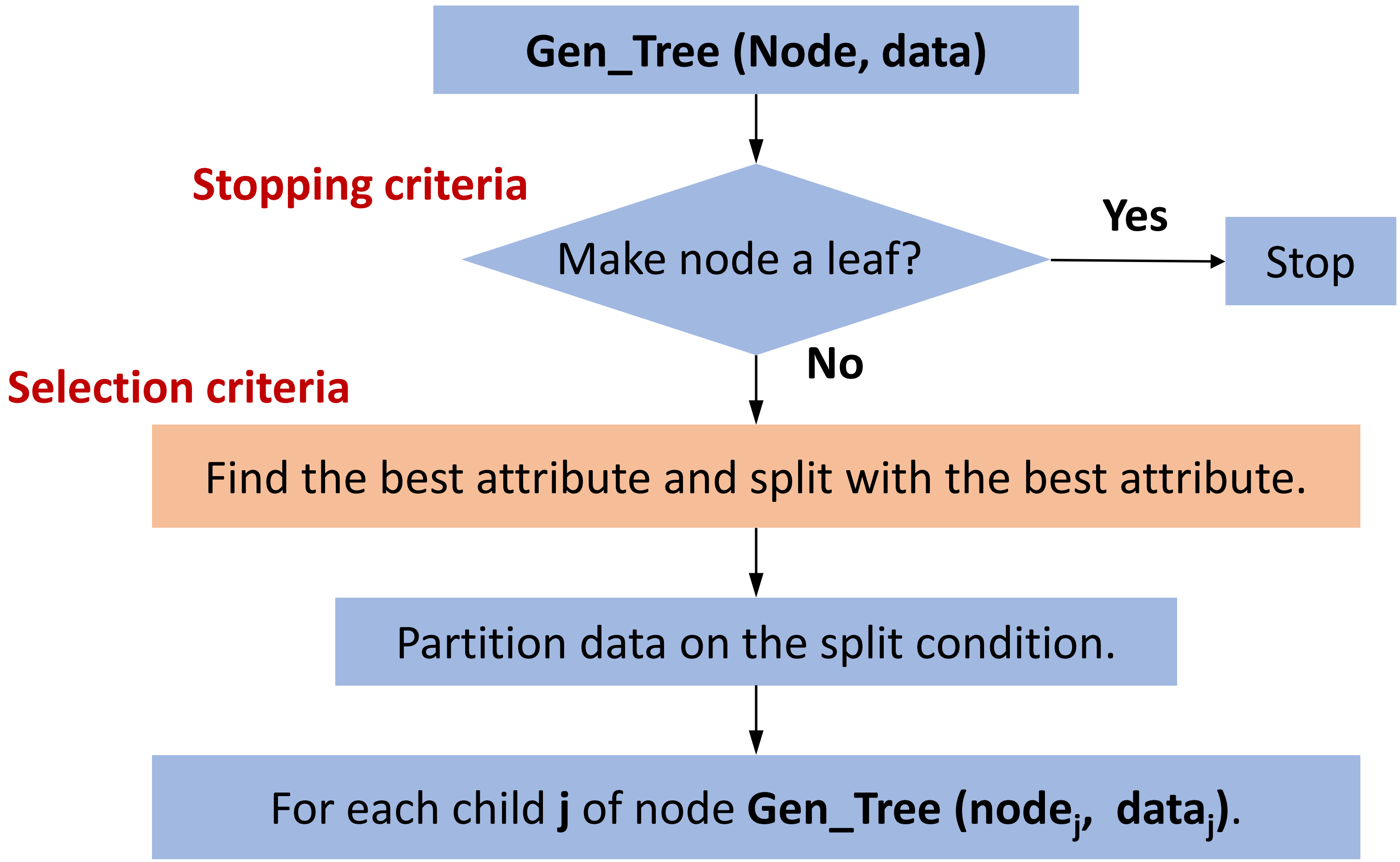
$$IG(Y \mid Cape) = H(Y) - H(Y \mid Cape)$$
$$= I(3,3) - \left(\frac{2}{6} \times I(2,0) + \frac{4}{6} \times I(1,3)\right)$$

$$IG(Y \mid Cape) = H(Y) - H(Y \mid Tie)$$
$$= I(3,3) - \left(\frac{2}{6} \times I(1,1) + \frac{4}{6} \times I(2,2)\right)$$

Iterative Dichotomiser 3 (ID3)

Decision Trees (ID3)

- **Top-down and divide-and-conquer approach**



Stopping Condition

- **All samples for a given node belong to the same class.**

- **There are no samples left.**

- **There are no attributes for additional partitioning.**
  - **The majority voting** is used to determine the leaf node.

Training Procedure

➢ **Selecting the attribute with the highest information gain**

$$H(Y) = I(3,3) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.0$$

$$H(Y \mid Gender) = \frac{5}{6}I(3,2) + \frac{1}{6}I(0,1) = \frac{5}{6}(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}) = 0.809$$

$$IG(Y \mid Gender) = 1.0 - 0.809 = 0.191$$

Training data

	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batman	Male	Yes	Yes	No	Yes	No	**Good**
Robin	Male	Yes	Yes	No	No	No	**Good**
Alfred	Male	No	No	Yes	No	No	**Good**
Penguin	Male	No	No	Yes	No	Yes	**Bad**
Catwoman	Female	Yes	No	No	Yes	No	**Bad**
Joker	Male	No	No	No	No	No	**Bad**

Training Procedure

➢ **Selecting the attribute with the highest information gain**

$$H(Y) = I(3,3) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.0$$

$$H(Y \mid Gender) = \frac{5}{6}I(3,2) + \frac{1}{6}I(0,1) = \frac{5}{6}(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}) = 0.809$$

$$H(Y \mid Mask) = \frac{3}{6}I(2,1) + \frac{3}{6}I(1,2) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

$$H(Y \mid Cape) = \frac{2}{6}I(2,0) + \frac{4}{6}I(1,3) = \frac{4}{6}(-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}) = 0.540$$

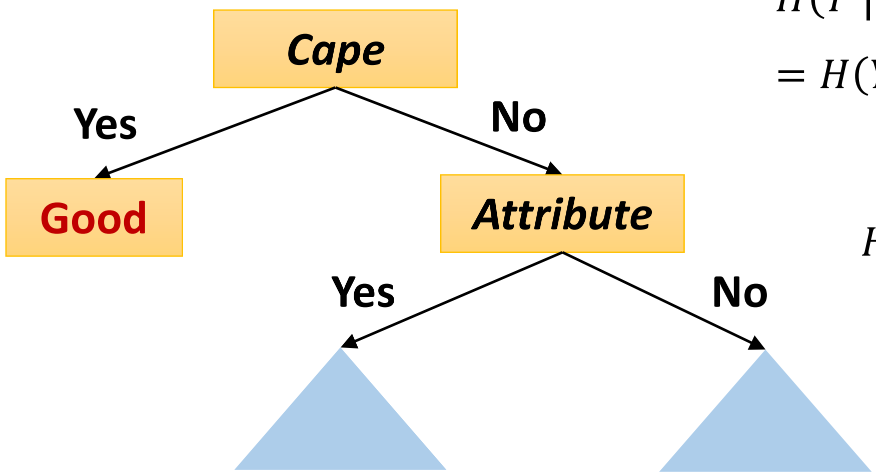
$$H(Y \mid Tie) = \frac{2}{6}I(1,1) + \frac{4}{6}I(2,2) = 1.0$$

$$H(Y \mid Ears) = \frac{2}{6}I(1,1) + \frac{4}{6}I(2,2) = 1.0$$

$$H(Y \mid Smokes) = \frac{1}{6}I(0,1) + \frac{5}{6}I(3,2) = 0.809$$

Training Procedure

- For the subset ($Cape = no$), select the attribute with the highest information gain.



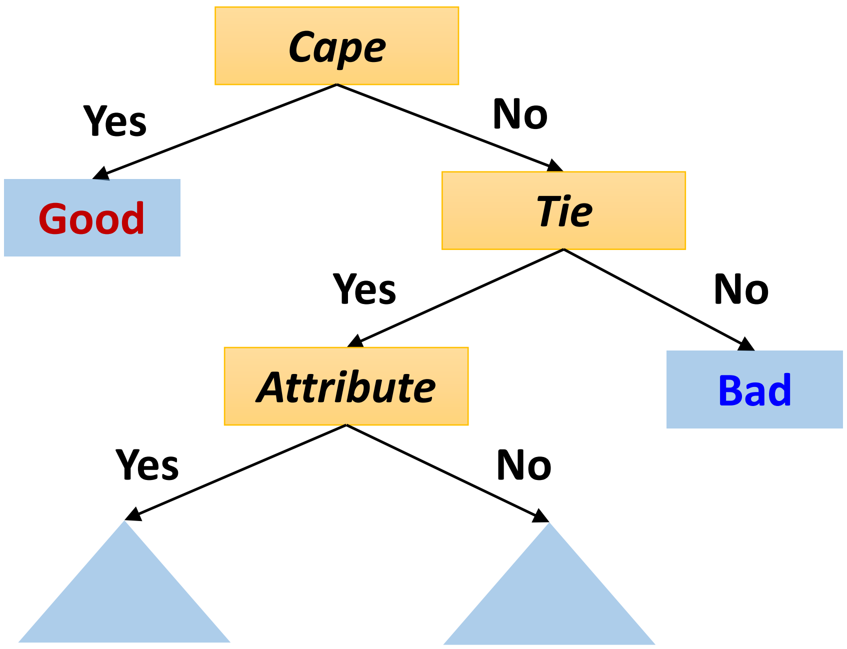
$$H(Y \mid Gender) = H(Y \mid Mask) = H(Y \mid Ears)$$
$$= H(Y \mid Smokes) = \frac{3}{4} I(1,2) + \frac{1}{4} I(0,1) = 0.689$$

$$H(Y \mid Tie) = \frac{2}{4} I(1,1) + \frac{2}{4} I(0,2) = 0.5$$

	Gender	*Mask*	*Tie*	*Ears*	*Smokes*	*Label*
Alfred	Male	No	Yes	No	No	**Good**
Penguin	Male	No	Yes	No	Yes	**Bad**
Catwoman	Female	Yes	No	Yes	No	**Bad**
Joker	Male	No	No	No	No	**Bad**

Training Procedure

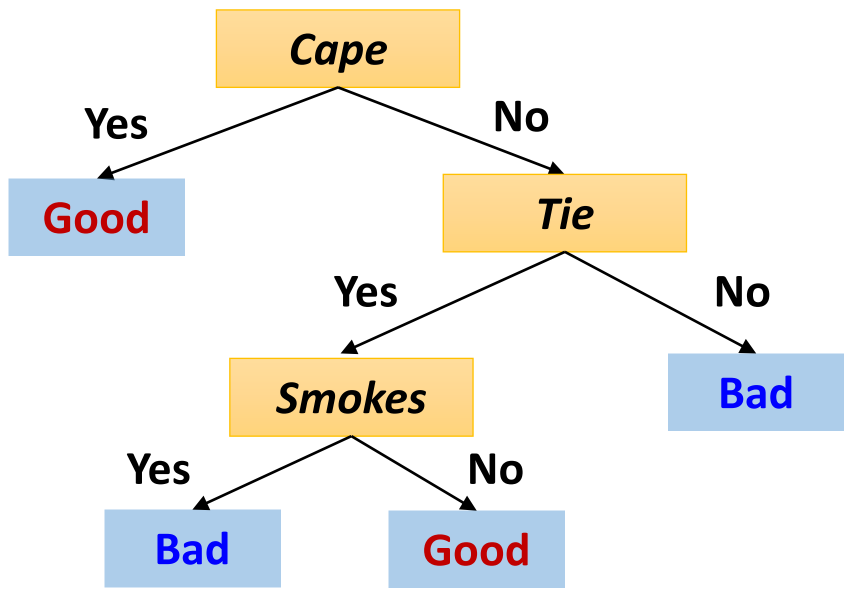
- For the subset ($Tie = yes$), select the attribute with the highest information gain.



	Gender	*Mask*	*Ears*	*Smokes*	*Label*
Alfred	Male	No	No	No	Good
Penguin	Male	No	No	Yes	Bad

Making Prediction

- Predicting the label for each sample on test data



Testing data

	Gender	*Mask*	*Cape*	*Tie*	*Ears*	*Smokes*	*Label*
Batgirl	Female	Yes	Yes	Yes	Yes	No	??
Riddler	Male	Yes	No	No	No	No	??

Example: Buying Laptops

Training data

Age	*Income*	*Student*	*Credit*	*Buy*
<= 30	High	N	Fair	**No**
<= 30	High	N	Excellent	**No**
31 … 40	High	N	Fair	**Yes**
> 40	Medium	N	Fair	**Yes**
> 40	Low	Y	Fair	**Yes**
> 40	Low	Y	Excellent	**No**
31 … 40	Low	Y	Excellent	**Yes**
<= 30	Medium	N	Fair	**No**
<= 30	Low	Y	Fair	**Yes**
> 40	Medium	Y	Fair	**Yes**
<= 30	Medium	Y	Excellent	**Yes**
31 … 40	Medium	N	Excellent	**Yes**
31 … 40	High	Y	Fair	**Yes**
> 40	Medium	N	Excellent	**No**

Example: Buying Laptops

- **Selecting the attribute with the highest information gain**

Age	p	n	$I(p,n)$
<= 30	2	3	0.971
30 … 40	4	0	0
> 40	3	2	0.971

$$H(Y) - H(Y \mid Age)$$
$$= H(Y) - \left(\frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(2,3)\right) =$$
$$= 0.940 - 0.69 = 0.25$$

Age	*Buy*
<= 30	No
<= 30	No
31 … 40	Yes
> 40	Yes
> 40	Yes
> 40	No
31 … 40	Yes
<= 30	No
<= 30	Yes
> 40	Yes
<= 30	Yes
31 … 40	Yes
31 … 40	Yes
> 40	No

Example: Buying Laptops

- Selecting the attribute with the highest information gain

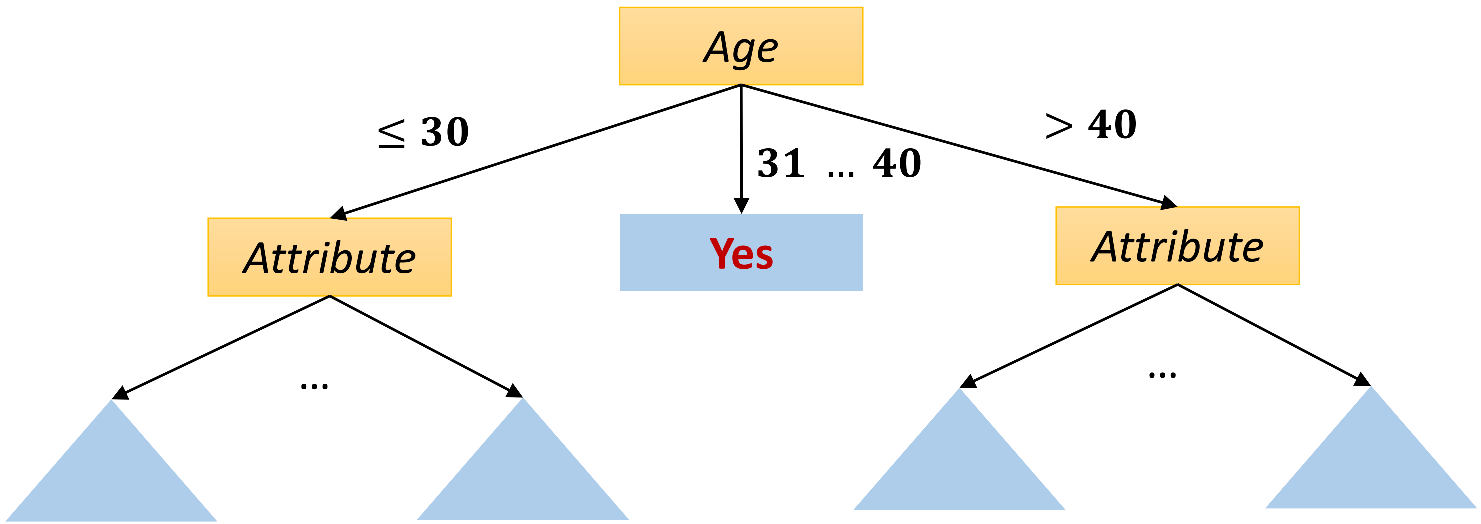
Student	p	n	$I(p,n)$
Yes	6	1	0.971
No	4	3	0

$$H(Y) - H(Y \mid Student)$$
$$= H(Y) - \left(\frac{7}{14} I(6{,}1) + \frac{7}{14} I(4{,}3)\right) = 0.151$$

Student	*Buy*
N	No
N	No
N	Yes
N	Yes
Y	Yes
Y	No
Y	Yes
N	No
Y	Yes
Y	Yes
Y	Yes
N	Yes
Y	Yes
N	No

Example: Buying Laptops

➢ A decision tree after the first partitioning



Income	*Student*	*Credit*	*Buy*
High	N	Fair	No
High	N	Excellent	No
Medium	N	Fair	No
Low	Y	Fair	Yes
Medium	Y	Excellent	Yes

Income	*Student*	*Credit*	*Buy*
Medium	N	Fair	Yes
Low	Y	Fair	Yes
Low	Y	Excellent	No
Medium	Y	Fair	Yes
Medium	N	Excellent	No

Example: Buying Laptops

- **Using age, it is partitioned into three subsets.**
  - For each subset, select the attribute with the highest information gain in a recursive manner.

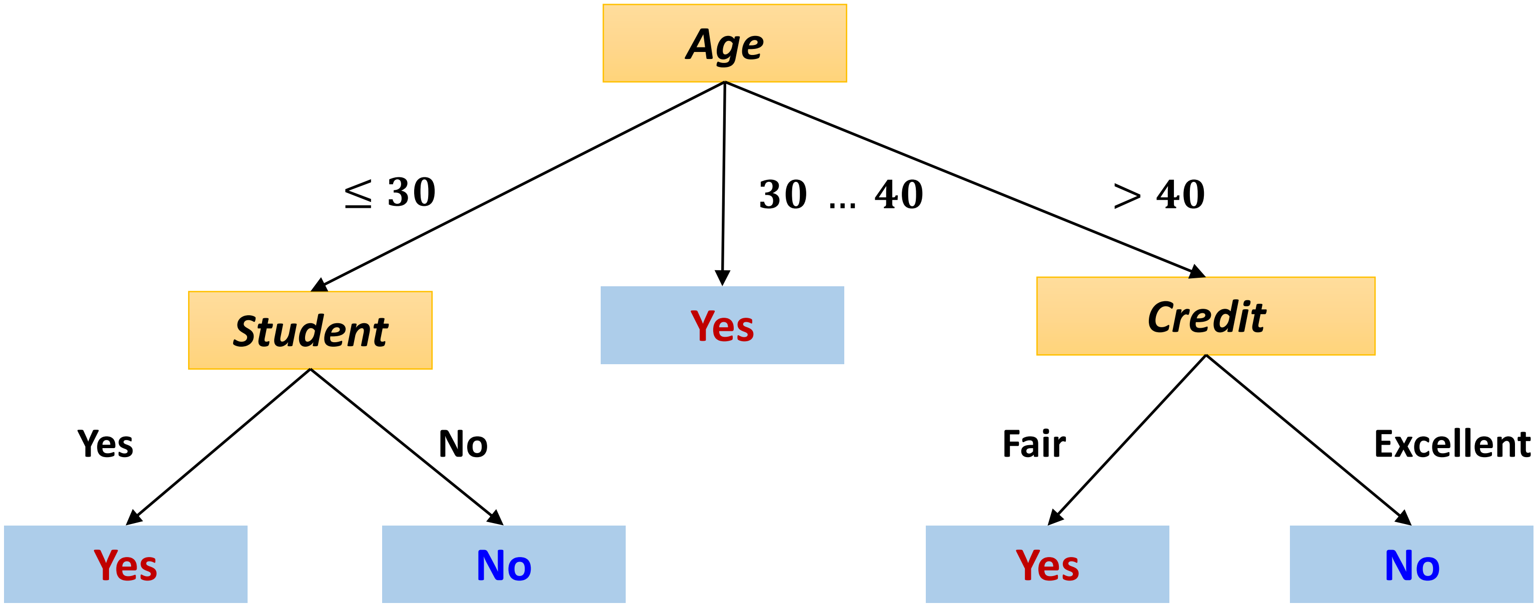
Income	Student	Credit	Buy
High	N	Fair	No
High	N	Excellent	No
Medium	N	Fair	No
Low	Y	Fair	Yes
Medium	Y	Excellent	Yes

Income	Student	Credit	Buy
High	N	Fair	Yes
Low	Y	Excellent	Yes
Medium	N	Excellent	Yes
High	Y	Fair	Yse

Income	Student	Credit	Buy
Medium	N	Fair	Yes
Low	Y	Fair	Yes
Low	Y	Excellent	No
Medium	Y	Fair	Yes
Medium	N	Excellent	No

- **Let each subset $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$.**
  - $\mathcal{D}_2$ has already been decided as a single label.

Example: Buying Laptops



C4.5: Improving ID3

C4.5: Improving ID3

- C4.5 has several improvements to ID3.
- Handling both continuous and discrete attributes
- Handling attributes with differing costs
- Pruning trees after creation

Handling Continuous Attributes

Training data

Outlook	*Temperature*	*Humidity*	*Wind*	*Play*
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	78	False	Yes
Rainy	70	96	False	Yes
Rainy	68	80	False	Yes
Rainy	65	70	True	No
Overcast	64	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	80	False	Yes
Sunny	75	70	True	Yse
Overcast	72	90	True	Yse
Overcast	81	75	False	Yse
Rainy	71	80	True	No

Handling Continuous Attributes

➢ **Finding the best split point by enumerating all possible cases**

Temp	*Humidity*	*Wind*	*Play*
85	85	False	No
80	90	True	No
83	78	False	Yes
70	96	False	Yes
68	80	False	Yes
65	70	True	No
64	65	True	Yes
72	95	False	No
69	70	False	Yes
75	80	False	Yes
75	70	True	Yse
72	90	True	Yse
81	75	False	Yse
71	80	True	No

$(1)\ \leq\ 64:\ [1+,0-], > 64:\ [8+,5-]$

$(2)\ \leq\ 65:\ [1+,1-], > 65:\ [8+,4-]$

$(3)\ \leq\ 68:\ [2+,1-], > 68:\ [7+,4-]$

$(4)\ \leq\ 69:\ [3+,1-], > 69:\ [6+,4-]$

$(5)\ \leq\ 70:\ [4+,1-], > 70:\ [5+,4-]$

$(6)\ \leq\ 71:\ [4+,2-], > 71:\ [5+,3-]$

$(7)\ \leq\ 72:\ [5+,3-], > 72:\ [4+,2-]$

$(8)\ \leq\ 75:\ [7+,3-], > 75:\ [2+,2-]$

$(9)\ \leq\ 80:\ [7+,4-], > 80:\ [2+,1-]$

$(10)\ \leq\ 81:\ [8+,4-], > 81:\ [1+,1-]$

$(11)\ \leq\ 83:\ [9+,4-], > 83:\ [0+,1-]$

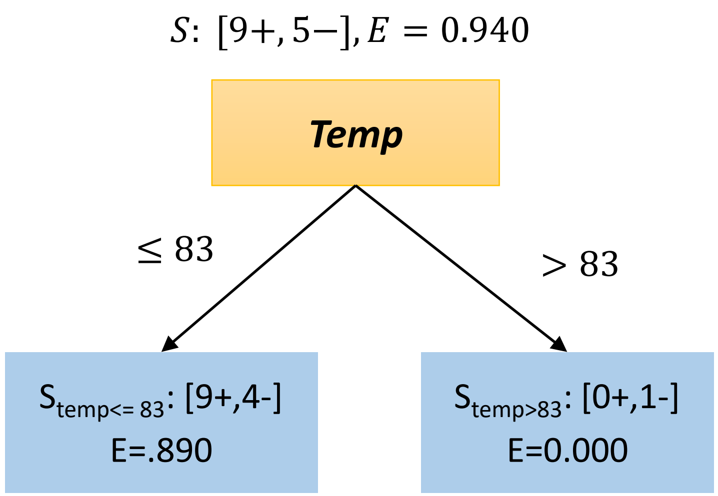
$(12)\ \leq\ 85:\ [9+,5-], > 85:\ [0+,0-]$

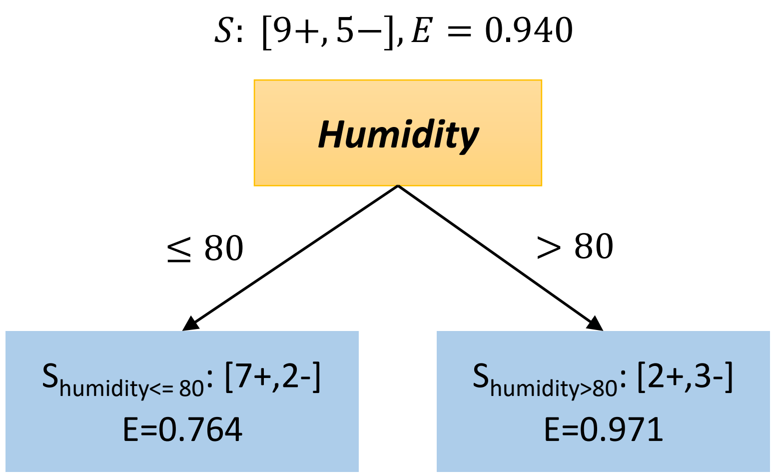
Handling Continuous Attributes

- **Finding the best split point by enumerating all possible cases**

$$\left.\begin{array}{l}
(1) \leq 64: [1+,0-], > 64: [8+,5-] \rightarrow Entropy = 0.893 \\
(2) \leq 65: [1+,1-], > 65: [8+,4-] \rightarrow Entropy = 0.930 \\
(3) \leq 68: [2+,1-], > 68: [7+,4-] \rightarrow Entropy = 0.940 \\
(4) \leq 69: [3+,1-], > 69: [6+,4-] \rightarrow Entropy = 0.925 \\
(5) \leq 70: [4+,1-], > 70: [5+,4-] \rightarrow Entropy = 0.895 \\
(6) \leq 71: [4+,2-], > 71: [5+,3-] \rightarrow Entropy = 0.939 \\
(7) \leq 72: [5+,3-], > 72: [4+,2-] \rightarrow Entropy = 0.939 \\
(8) \leq 75: [7+,3-], > 75: [2+,2-] \rightarrow Entropy = 0.915 \\
(9) \leq 80: [7+,4-], > 80: [2+,1-] \rightarrow Entropy = 0.940 \\
(10) \leq 81: [8+,4-], > 81: [1+,1-] \rightarrow Entropy = 0.930 \\
(11) \leq 83: [9+,4-], > 83: [0+,1-] \rightarrow Entropy = 0.827 \\
(12) \leq 85: [9+,5-], > 85: [0+,0-] \rightarrow Entropy = 0.940
\end{array}\right\} \min \longrightarrow 0.827$$

Handling Continuous Attributes





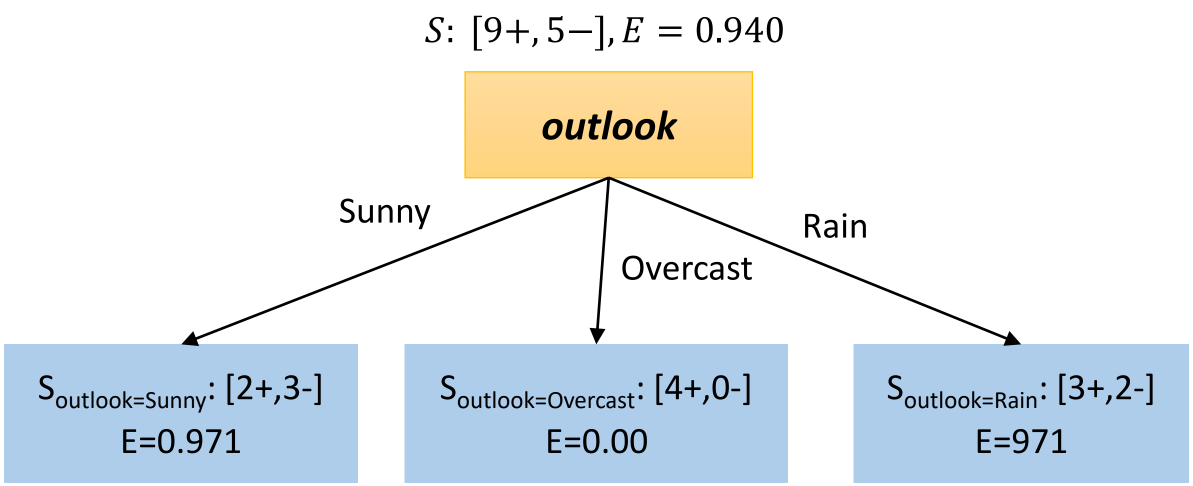
Gain(*temp*)

$$= .940 - (13/14) \times .890 - (1/14) \times 1.0 = .113$$

Gain(*humidity*)

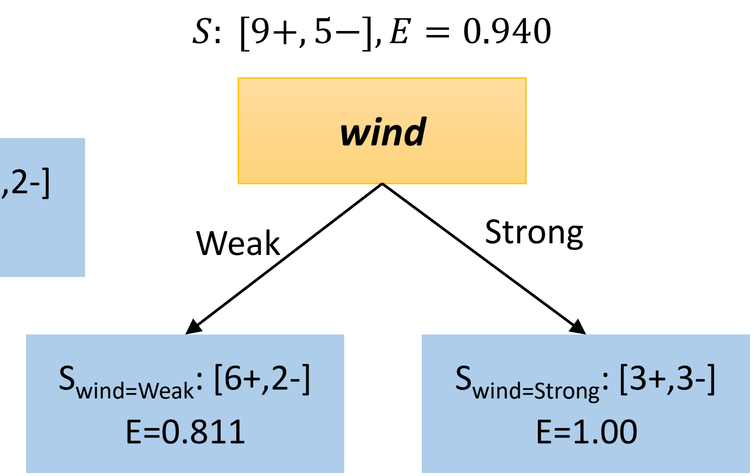
$$= .940 - (9/14) \times 0.764 - (5/14) \times .971 = .102$$

Handling Continuous Attributes



Gain(*outlook*)

$$= .940 - (5/14) \times .971 - (4/14) \times 1.0 - (5/14) \times .971 = .246$$



Gain(*wind*)

$$= .940 - (8/14) \times .811 - (6/14) \times 1.0 = .048$$

Problem of Information Gain

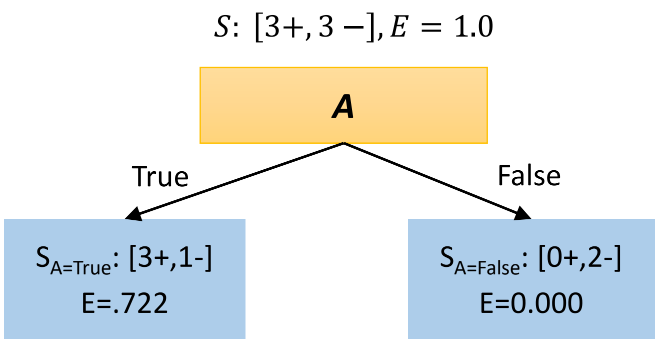
- Selecting the attribute with the highest information gain

A	*B*	*Play*
True	B1	Y
True	B2	Y
True	B3	Y
True	B4	N
False	B5	N
False	B6	N

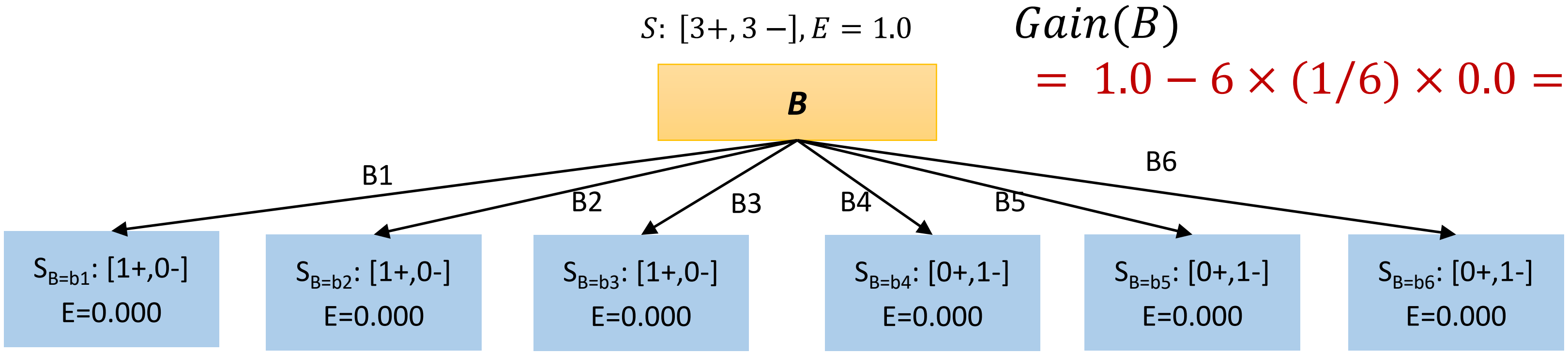
- Which attribute is better?

Problem of Information Gain

- Entropy does not consider a **branching factor.**



$$Gain(A) = 1.0 - (4/6) \times .722 - (2/6) \times 1.0 = .484$$



$$Gain(B) = 1.0 - 6 \times (1/6) \times 0.0 = 1.0$$

GainInfo: Alternative Measure

- **How to considering a branching factor?**

$$SplitInfo(Y) = -\sum_{i=1}^{k} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \log_2 \frac{|\mathcal{D}_i|}{|\mathcal{D}|}$$

$$\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_k\}$$

A	B	Play
True	B1	Y
True	B2	Y
True	B3	Y
True	B4	N
False	B5	N
False	B6	N

$$SplitInfo(A) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$$

$$SplitInfo(B) = 6\left(-\frac{1}{6}\log_2\frac{1}{6}\right) = 2.585$$

GainInfo: Alternative Measure

- How to considering a **branching factor?**

$$GainInfo(Y) = \frac{Gain(Y)}{SplitInfo(Y)}$$

A	B	Play
True	B1	Y
True	B2	Y
True	B3	Y
True	B4	N
False	B5	N
False	B6	N

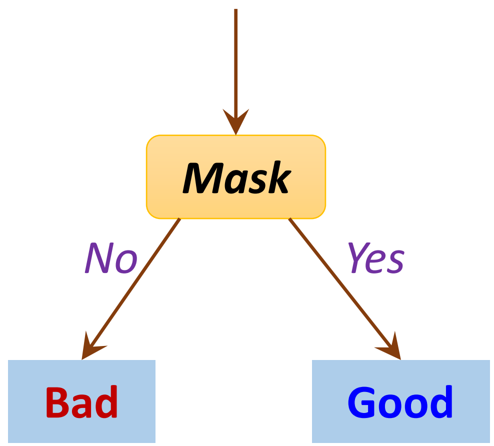
$$GainInfo(A) = \frac{0.484}{0.918} = 0.527$$

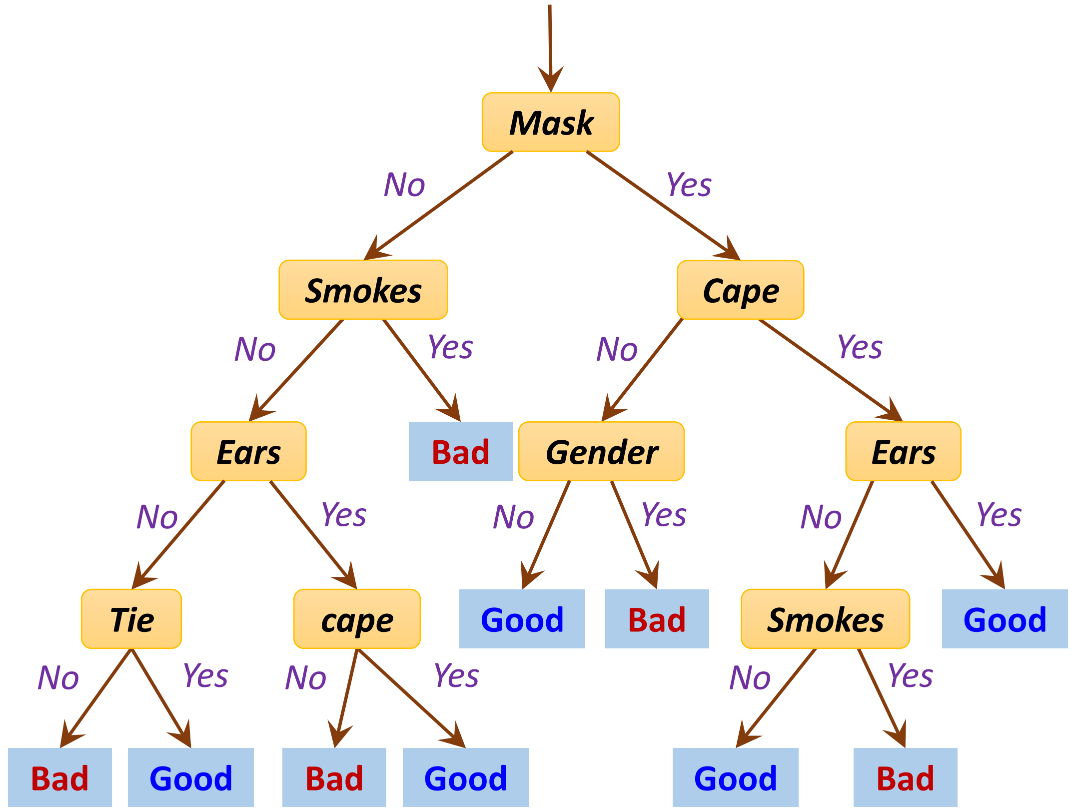
$$GainInfo(B) = \frac{1.000}{2.585} = 0.387$$

- Therefore, A is better than B.

Underfitting vs. Overfitting

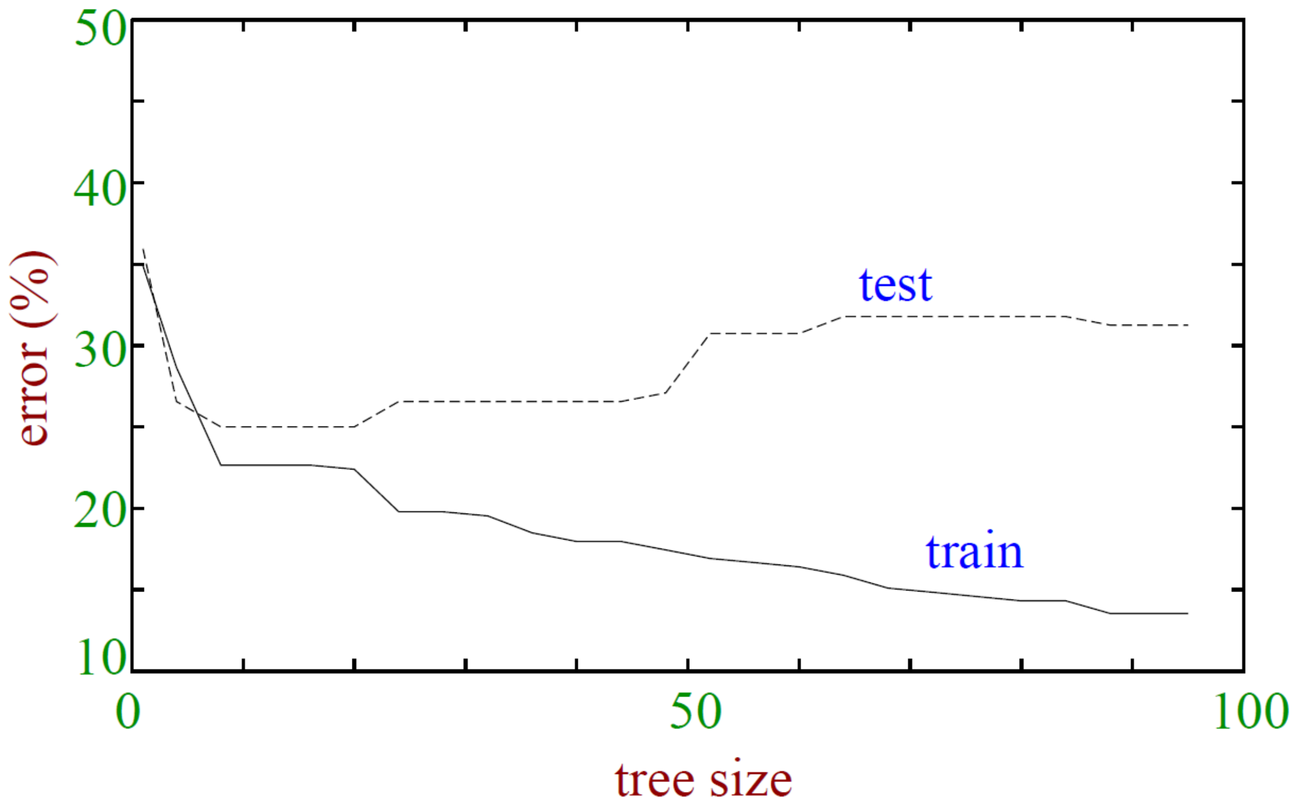
- Underfitting: overly simple, does not even fit available data
- Overfitting: too complicated, perfectly classifies training data





Tree Size vs. Accuracy

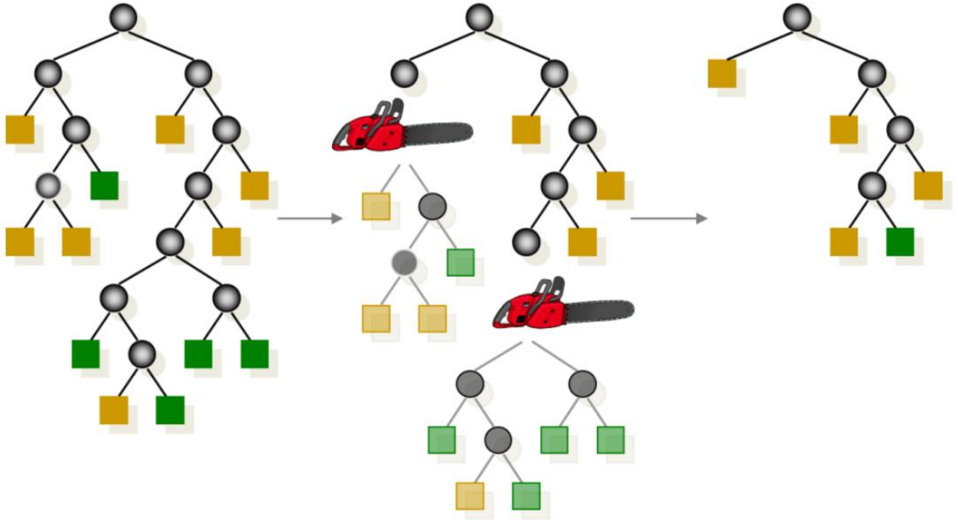
- **Trees must be big enough to fit training data.**
- **However, too big trees may overfit.**
  - It is difficult to decide the **best tree size** from training errors.



Two Solutions to Avoid Overfitting

- **Prepruning**
  - Do not split a node if this would result in **the goodness measure falling below a threshold.**
  - It is difficult to choose an **appropriate threshold**.

- **Postpruning**
  - Remove **branches** from a **fully grown tree**.



Q&A