# Decision Trees

Nipun Batra

Jan 8, 2019

# Training Data

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Example borrowed from Tom Mitchell's text book

# Training Data

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | [illegible] | [illegible] | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Input features

Example borrowed from Tom Mitchell's text book

# Training Data

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | [illegible] |
| D6 | Rain | [illegible] | [illegible] | Strong | [illegible] |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Input features

Output Variable

Example borrowed from Tom Mitchell's text book

# Training Data

Discrete Output : Classification

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | [illegible] | [illegible] | Strong | [illegible] |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Input features

Output Variable

Example borrowed from Tom Mitchell's text book

# Training Data

| Home # | Square footage | Age | Price (1000 USD) |
|---|---|---|---|
| D1 | 1000 | 20 | 30 |
| D2 | 2000 | 10 | No |
| D3 | 3000 | 5 | 40 |
| D4 | 1240 | 10 | 50 |

# Training Data

| Home # | Square footage | Age | Price (1000 USD) |
|---|---|---|---|
| D1 | 1000 | 20 | 30 |
| D2 | 200 | 10 | No |
| D3 | 3000 | 5 | 40 |
| D4 | 1240 | 10 | 50 |

Input

# Training Data

| Home # | Square footage | Age | Price (1000 USD) |
|---|---|---|---|
| D1 | 1000 | 20 | 30 |
| D2 | 200 | 10 | |
| D3 | 3000 | 5 | 40 |
| D4 | 1240 | 10 | 50 |

Input

Output

# Training Data

Continuous Output : Regression

| Home # | Square footage | Age | Price (1000 USD) |
|---|---|---|---|
| D1 | 1000 | 20 | 30 |
| D2 | 200 | 10 | |
| D3 | 3000 | 5 | 40 |
| D4 | 1240 | 10 | 50 |

Input

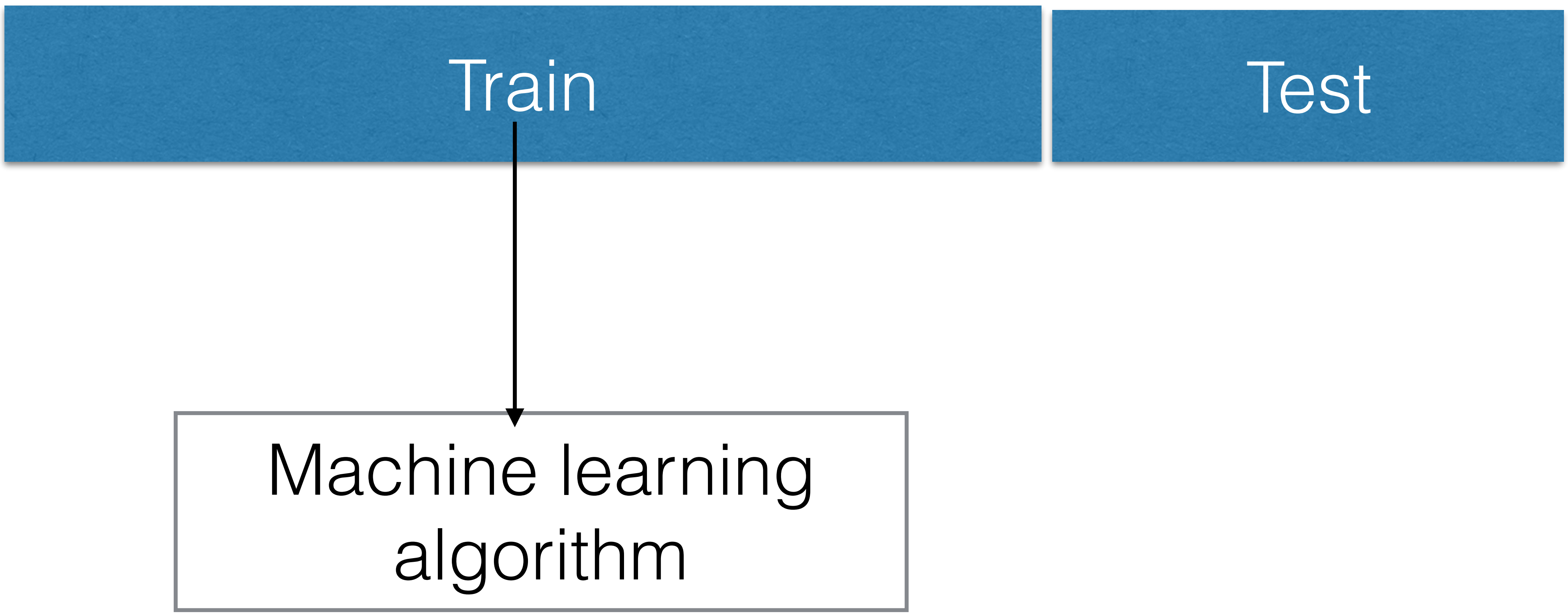
Output

# Train, Validation, Test

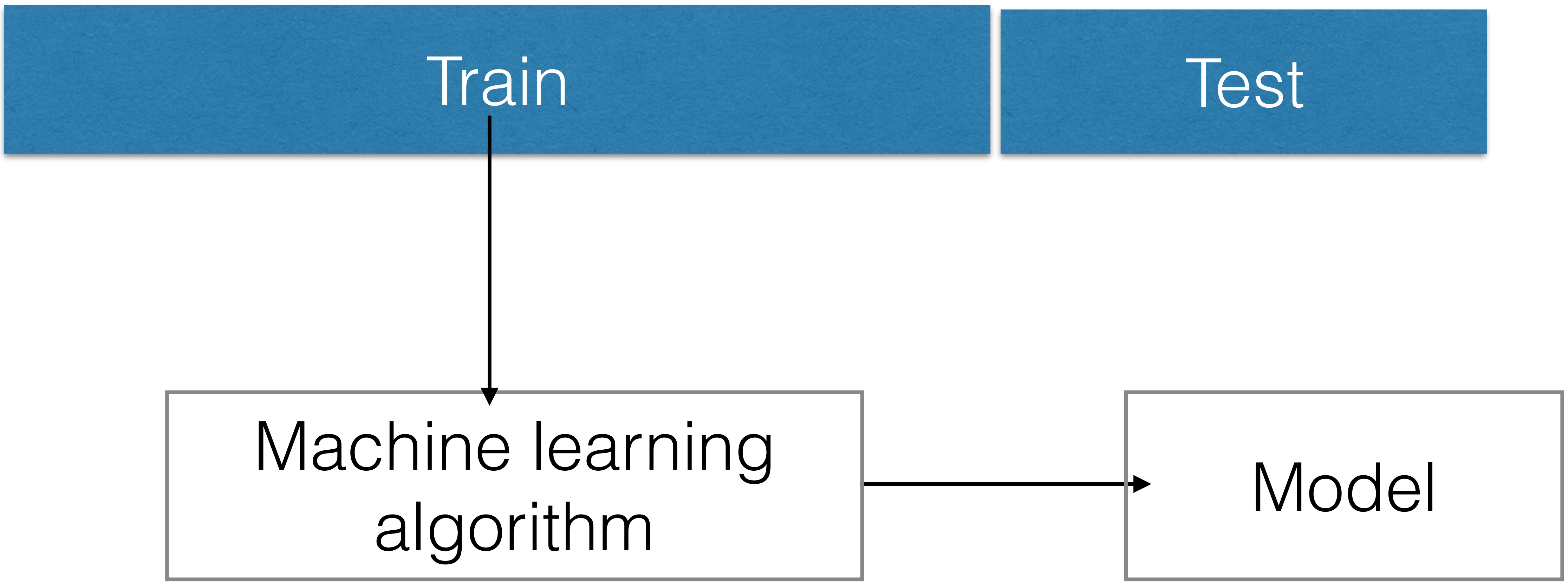
Data

# Train, Validation, Test

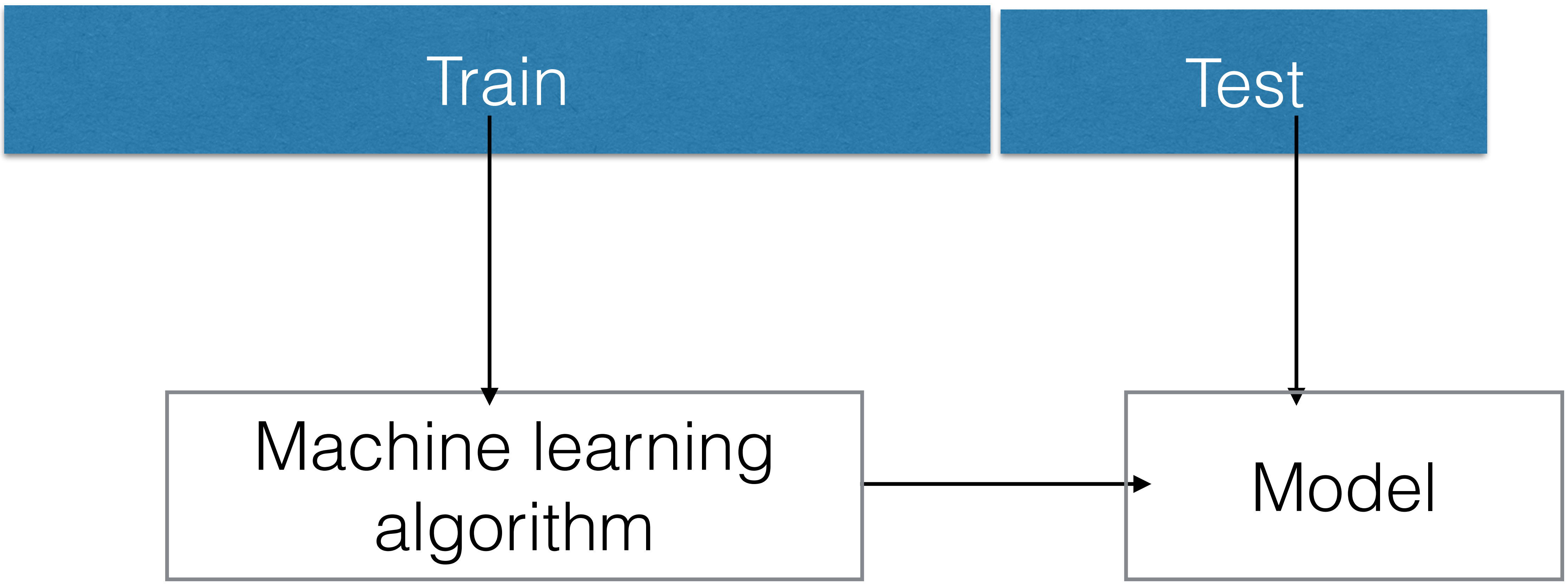
# Train, Validation, Test



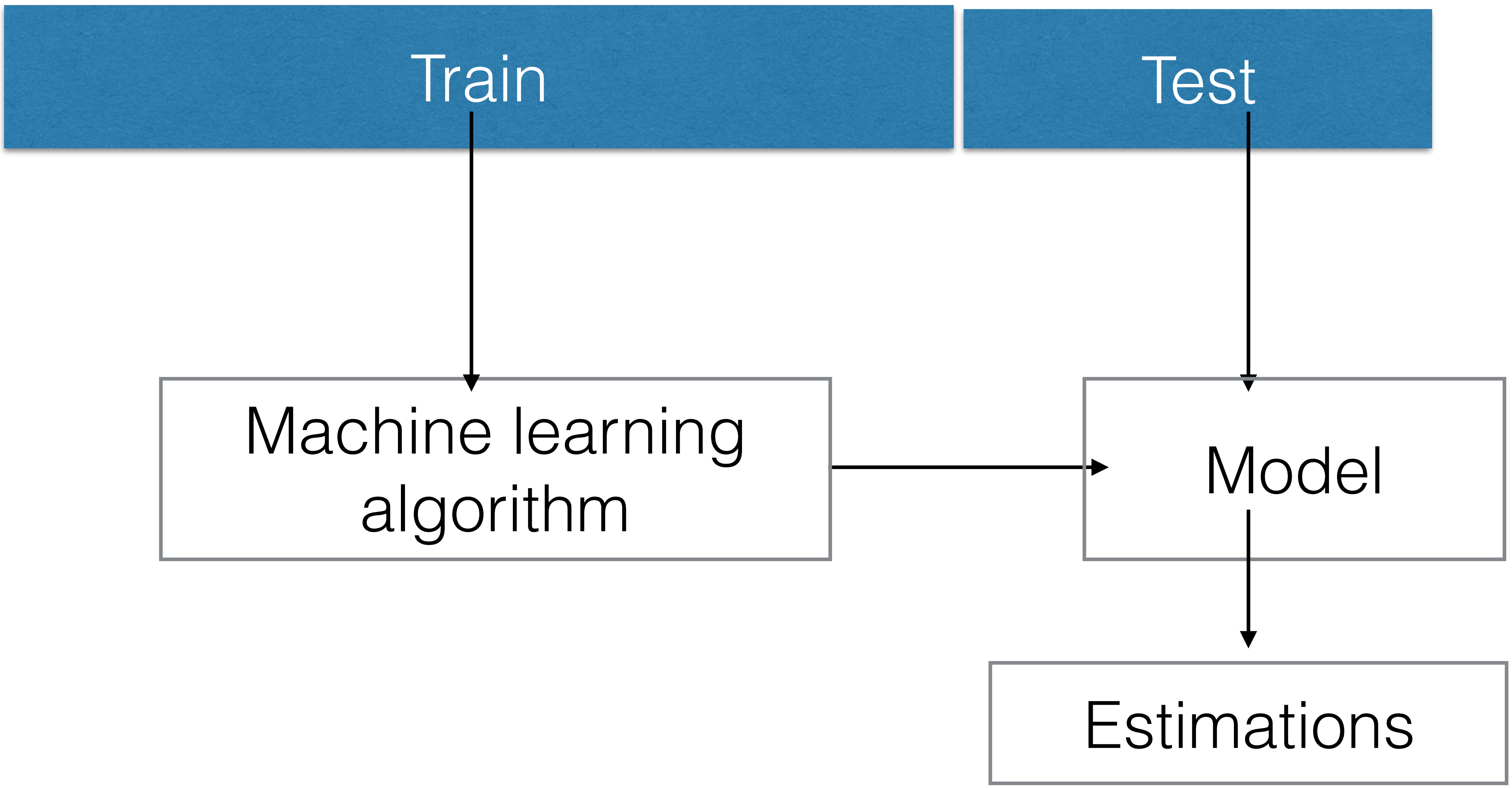
# Train, Validation, Test



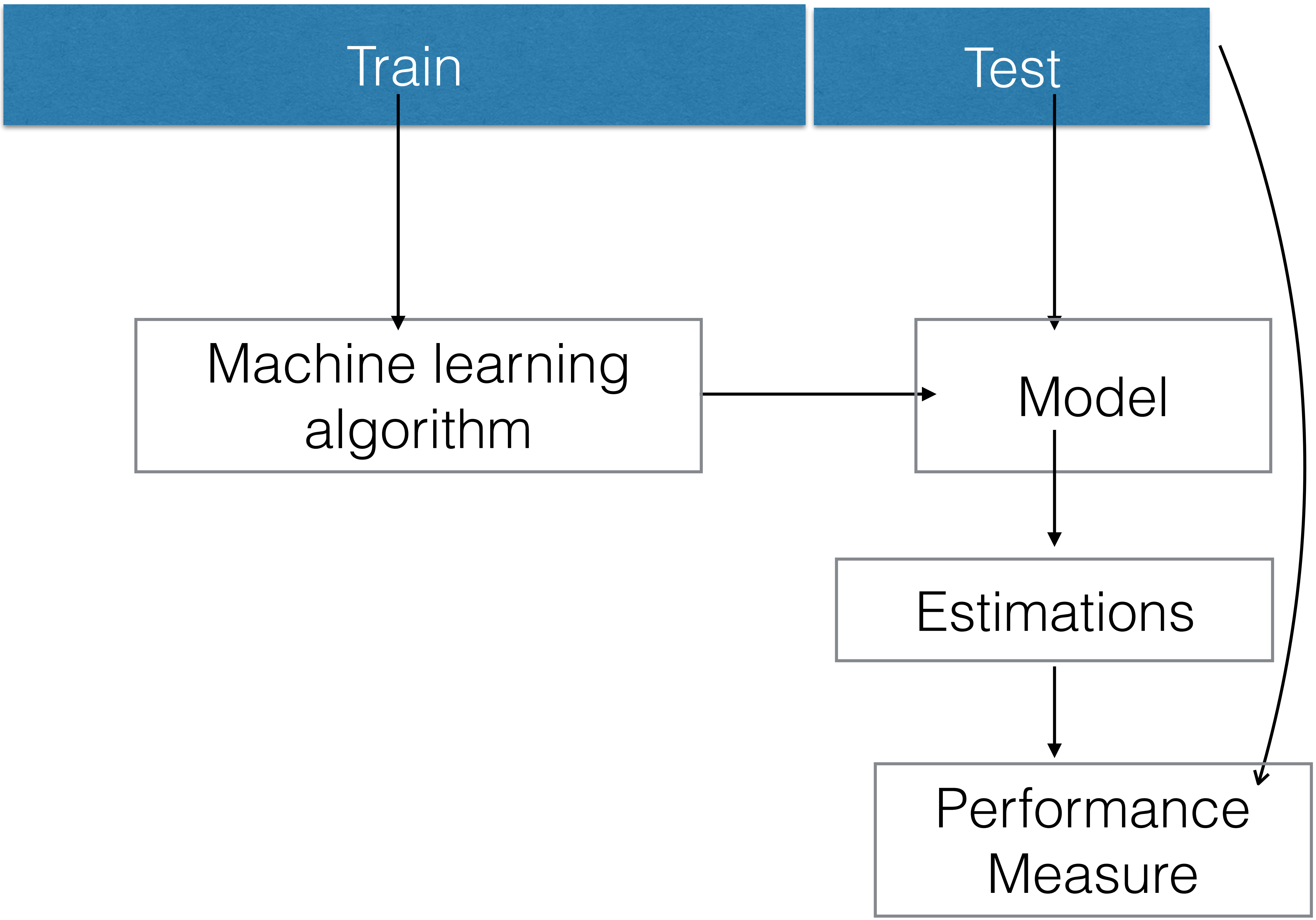
# Train, Validation, Test



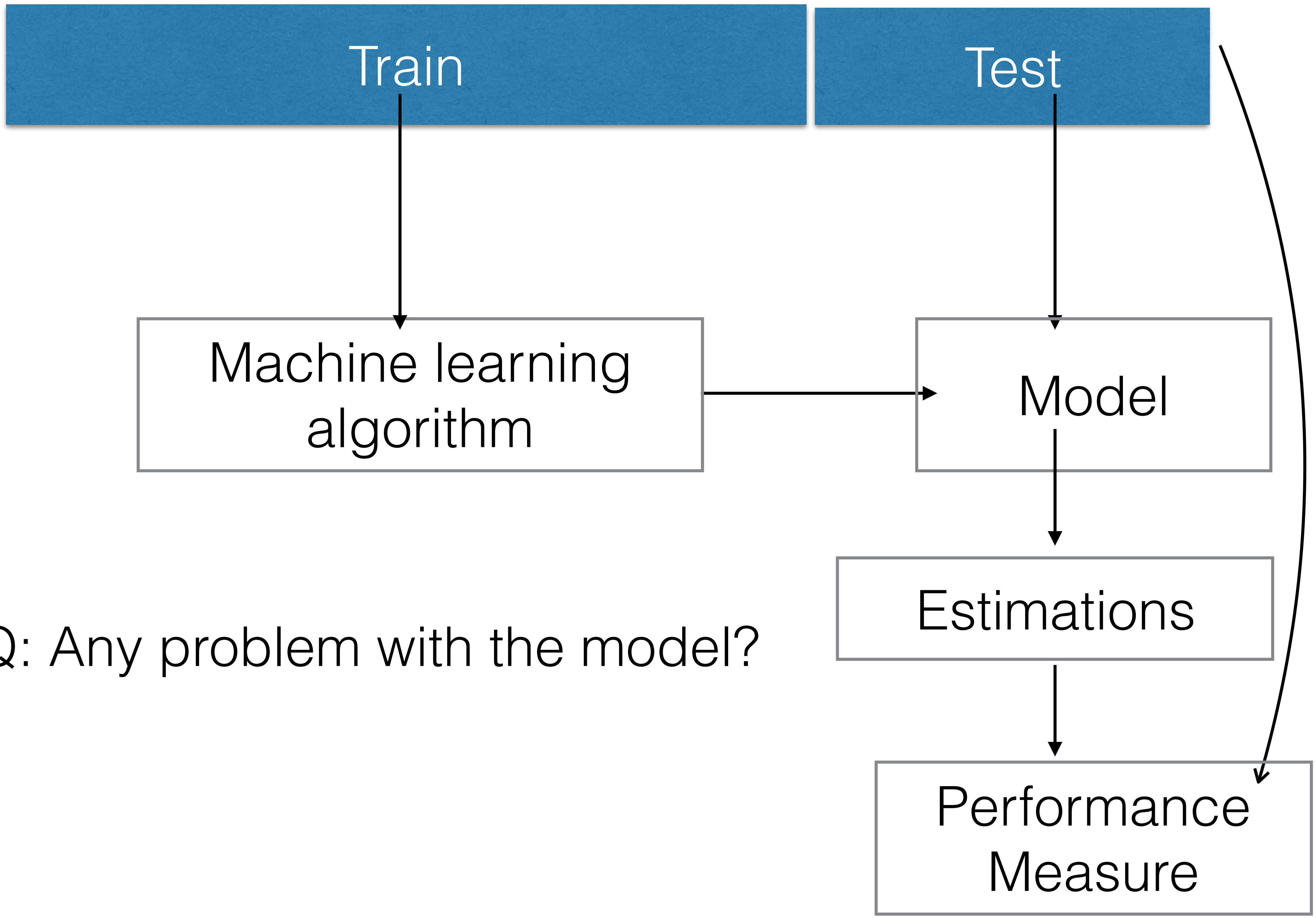
# Train, Validation, Test



# Train, Validation, Test

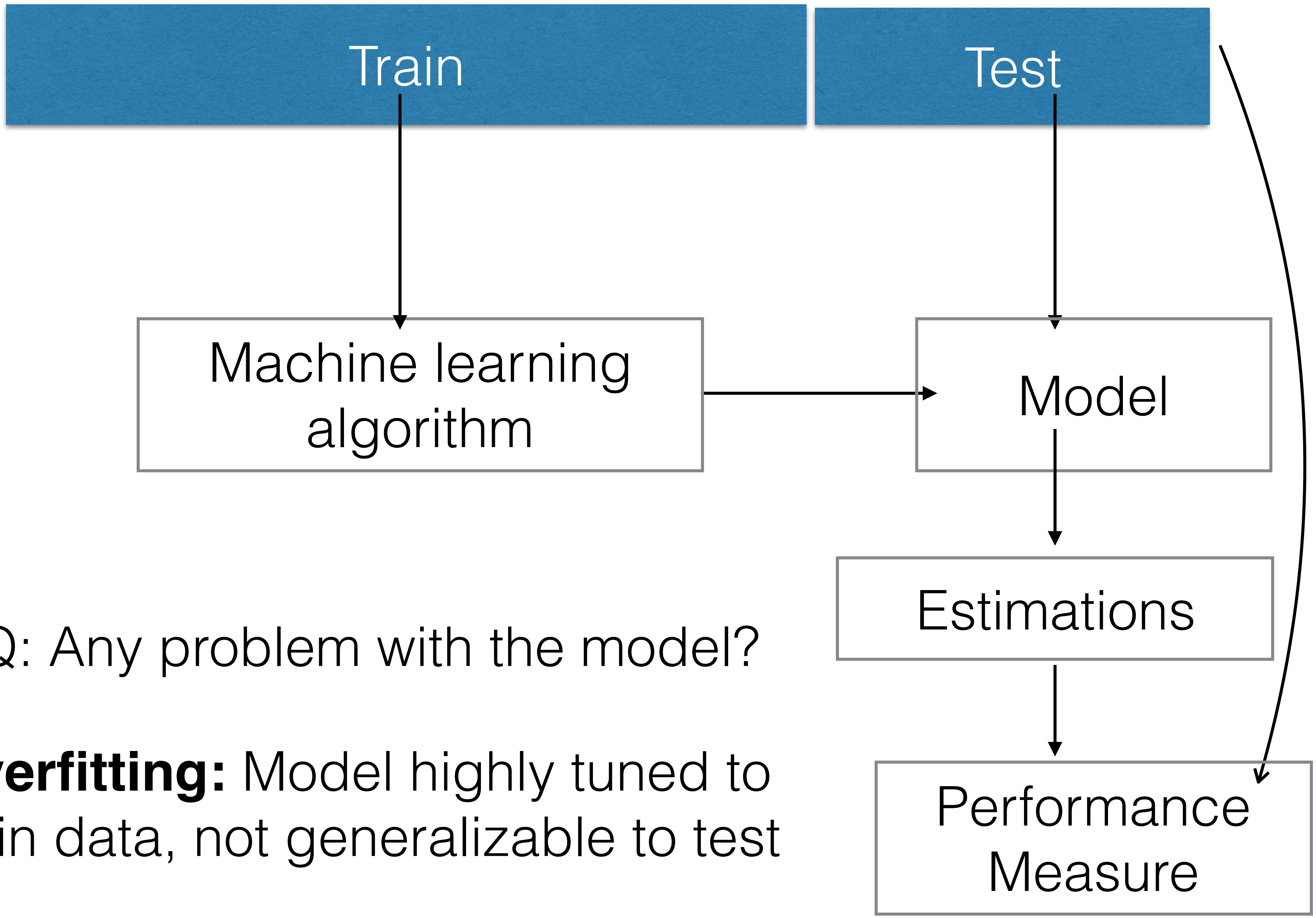


# Train, Validation, Test




Q: Any problem with the model?

# Train, Validation, Test




Q: Any problem with the model?

**Overfitting:** Model highly tuned to train data, not generalizable to test

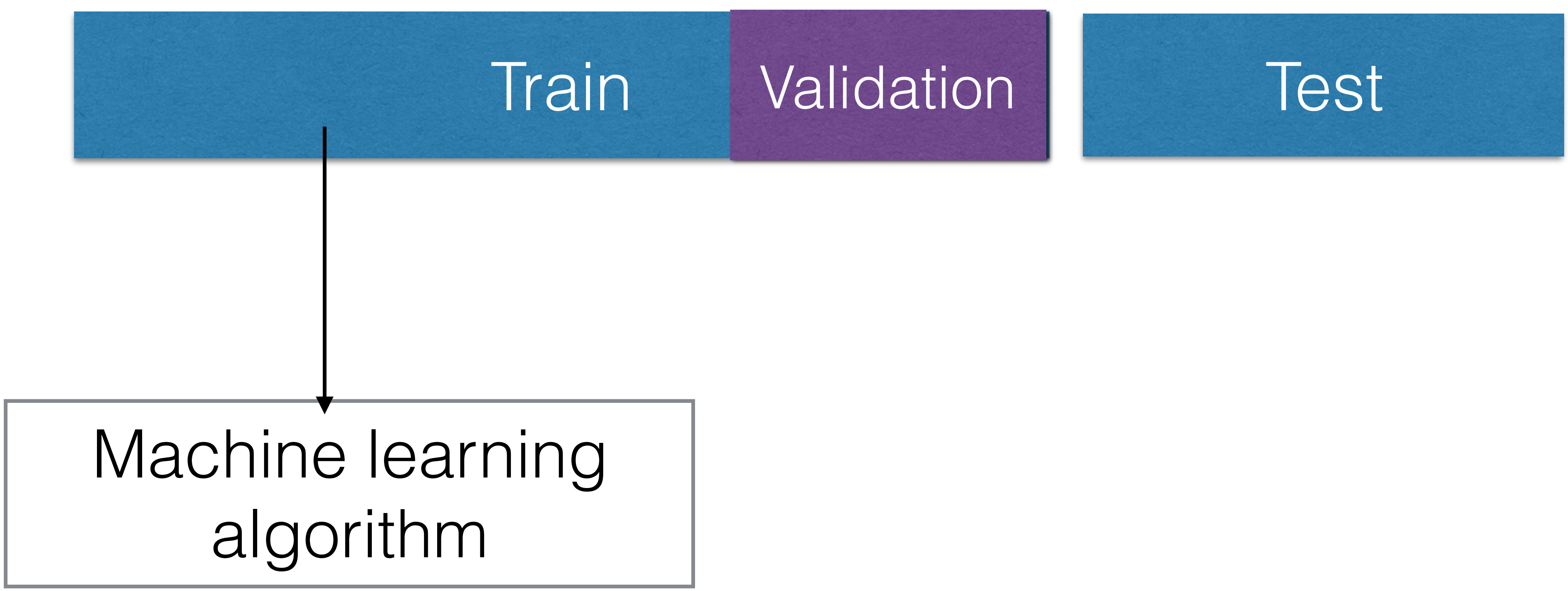
# Train, Validation, Test


Basic idea: Choose a model (parameters) which optimizes performance on validation set

# Train, Validation, Test

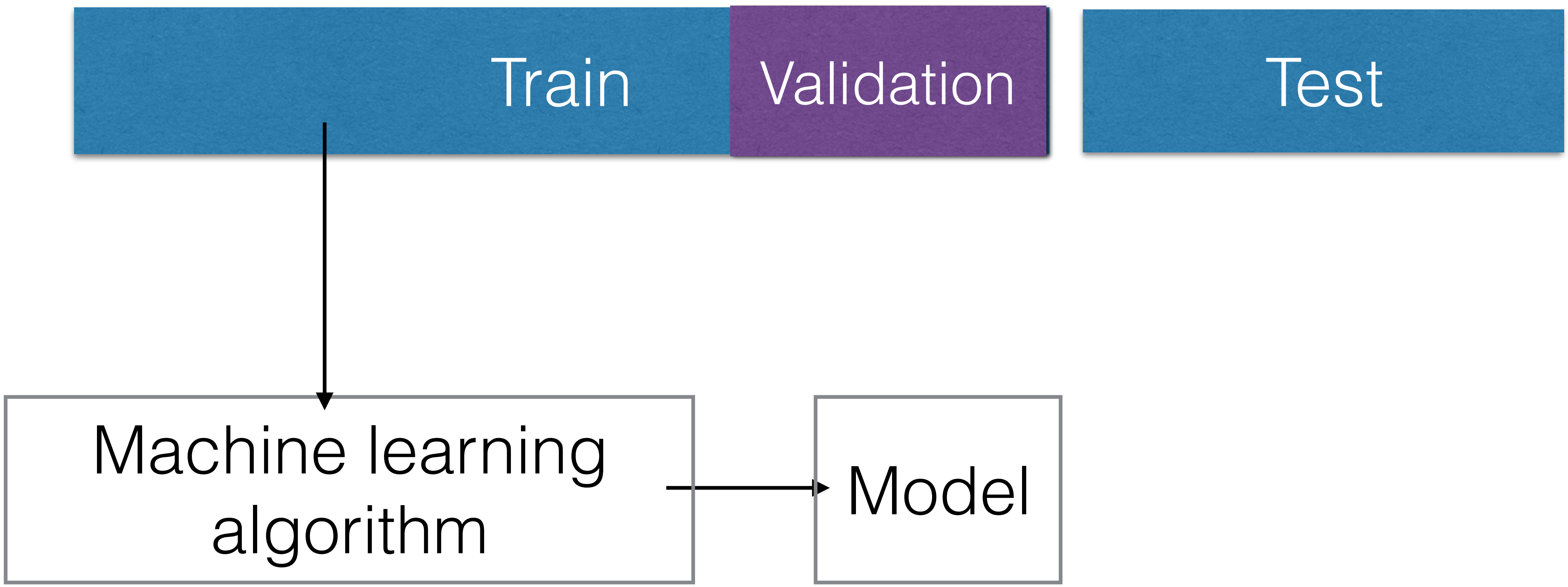

Basic idea: Choose a model (parameters) which optimizes performance on validation set

# Train, Validation, Test




Basic idea: Choose a model (parameters) which optimizes performance on validation set

# Train, Validation, Test




Basic idea: Choose a model (parameters) which optimizes performance on validation set

# Need For Interpretability

**How to Maintain Trust in AI**

Beyond developing initial trust, however, creators of AI also must work to maintain that trust. Siau and Wang suggest seven ways of "developing continuous trust" beyond the initial phases of product development:

- **Usability and reliability.** AI "should be designed to operate easily and intuitively," Siau and Wang write. "There should be no unexpected downtime or crashes."
- **Collaboration and communication.** AI developers want to create systems that perform autonomously, without human involvement. Developers must focus on creating AI applications that smoothly and easily collaborate and communicate with humans.
- **Sociability and bonding.** Building social activities into AI applications is one way to strengthen trust. A robotic dog that can recognize its owner and show affection is one example, Siau and Wang write.
- **Security and privacy protection.** AI applications rely on large data sets, so ensuring privacy and security will be crucial to establishing trust in the applications.
- **Interpretability.** Just as transparency is instrumental in building initial trust, interpretability – or the ability for a machine to explain its conclusions or actions – will help sustain trust.

# Training Data

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

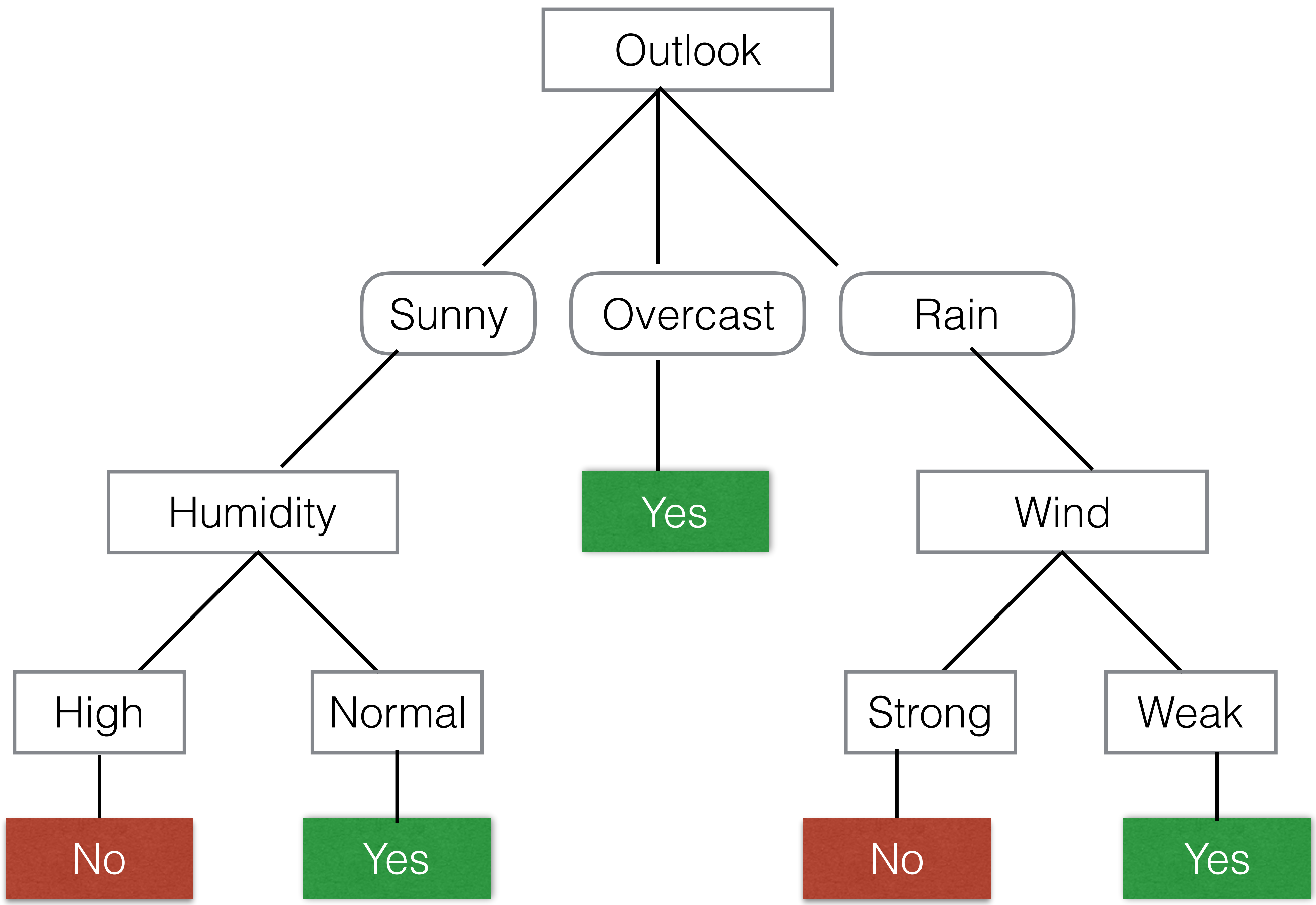
Example borrowed from Tom Mitchell's text book

# Training Data

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | High | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Example borrowed from Tom Mitchell's text book

# Decisions - Will I Play Tennis?




Example borrowed from Tom Mitchell's text book

# Hard to Learn Optimal Decision Tree

Volume 5, number 1 INFORMATION PROCESSING LETTERS May 1976

**CONSTRUCTING OPTIMAL BINARY DECISION TREES IS NP-COMPLETE***

Laurent HYAFIL
*IRIA – Laboria, 78150 Rocquencourt, France*

and

Ronald L. RIVEST
*Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, Massachusetts 02139, USA*

Received 7 November 1975, revised version received 26 January 1976

Binary decision trees, computational complexity, NP-complete

# Greedy Algorithm

Intuition:  At each level, choose an attribute that gives "biggest estimated performance gain"

# Greedy Algorithm

Intuition: At each level, choose an attribute that gives "biggest estimated performance gain"

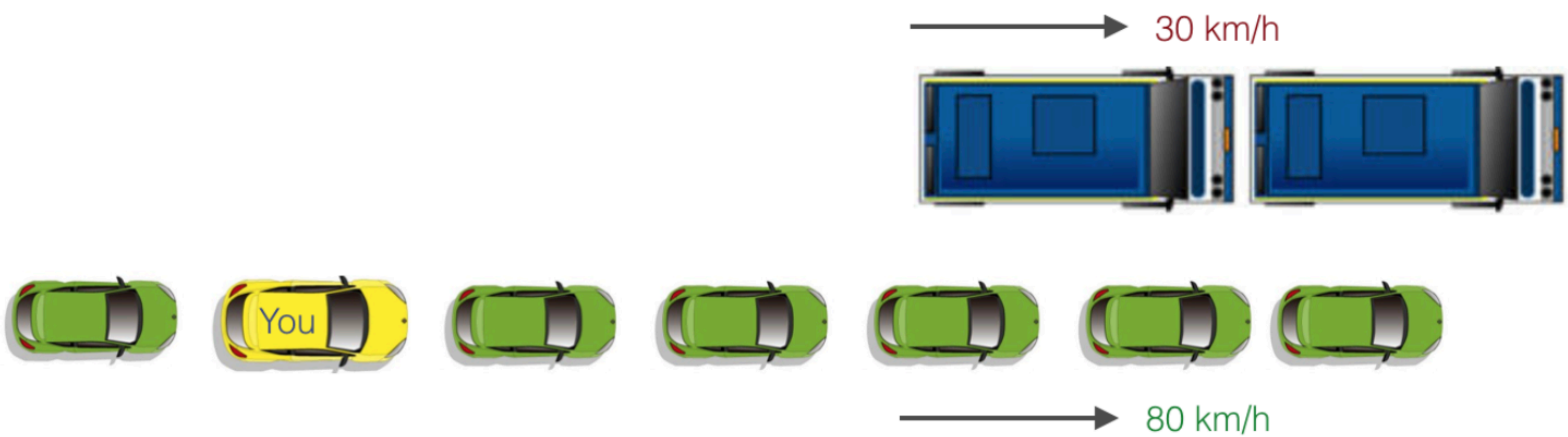



Image source: analyticsvidhya

# Greedy Algorithm

Intuition: At each level, choose an attribute that gives "biggest estimated performance gain"

Greedy! = Optimal

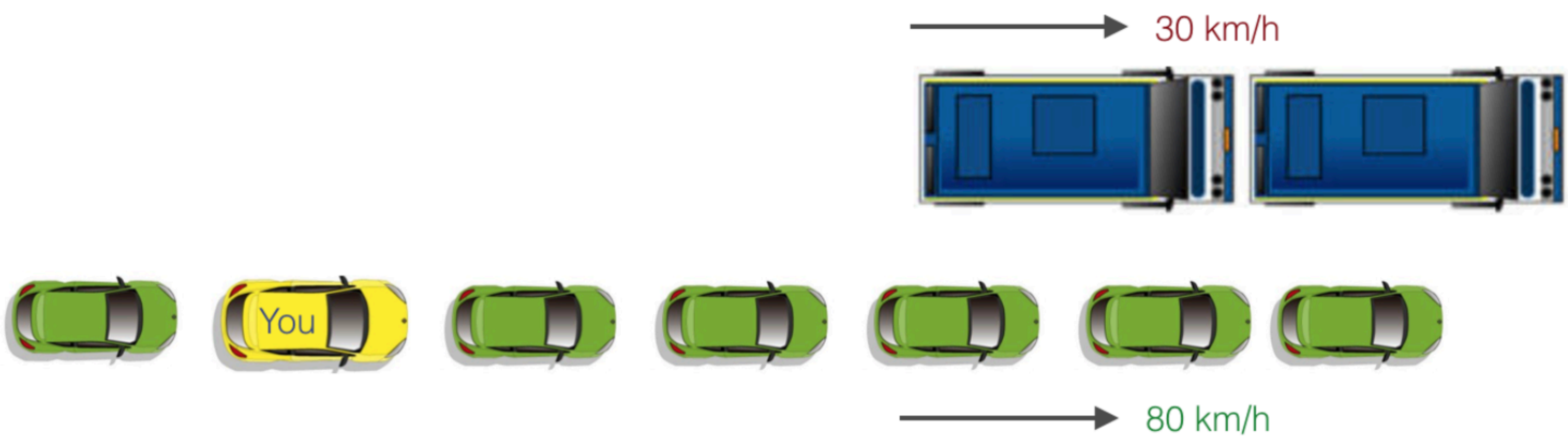



Image source: analyticsvidhya

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A
    3. For each value (v) of A

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A
    3. For each value (v) of A
        1. Add new tree branch : A = v

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A
    3. For each value (v) of A
        1. Add new tree branch : A = v
        2. Examples_v : subset of examples that A = v

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A
    3. For each value (v) of A
        1. Add new tree branch : A = v
        2. Examples_v : subset of examples that A = v
        3. If Examples_v is empty: add leaf with label = most common value of Target_Attribute

# Greedy Algorithm

ID3 (Examples, Target_Attribute, Attributes)

1. Create a root node for tree
2. If all examples are +/-, return root with label = +/-
3. If attributes = empty, return root with most common value of Target_Attribute in Examples
4. Begin
    1. A <- attribute from Attributes which **best** classifies Examples
    2. Root <- A
    3. For each value (v) of A
        1. Add new tree branch : A = v
        2. Examples_v : subset of examples that A = v
        3. If Examples_v is empty: add leaf with label = most common value of Target_Attribute
        4. Else: ID3 (Examples_v, Target_attribute, Attributes - {A})

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

| PlayTennis |
|---|
| No |
| No |
| Yes |
| Yes |
| Yes |
| No |
| Yes |
| No |
| Yes |
| Yes |
| Yes |
| Yes |
| Yes |
| No |

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

| PlayTennis |
|---|
| No |
| No |
| Yes |
| Yes |
| Yes |
| No |
| Yes |
| No |
| Yes |
| Yes |
| Yes |
| Yes |
| Yes |
| No |

5 No, 9 Yes

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

| PlayTennis |
|---|
| No |
| No |
| Yes |
| Yes |
| Yes |
| No |
| Yes |
| No |
| Yes |
| Yes |
| Yes |
| Yes |
| Yes |
| No |

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

# Entropy

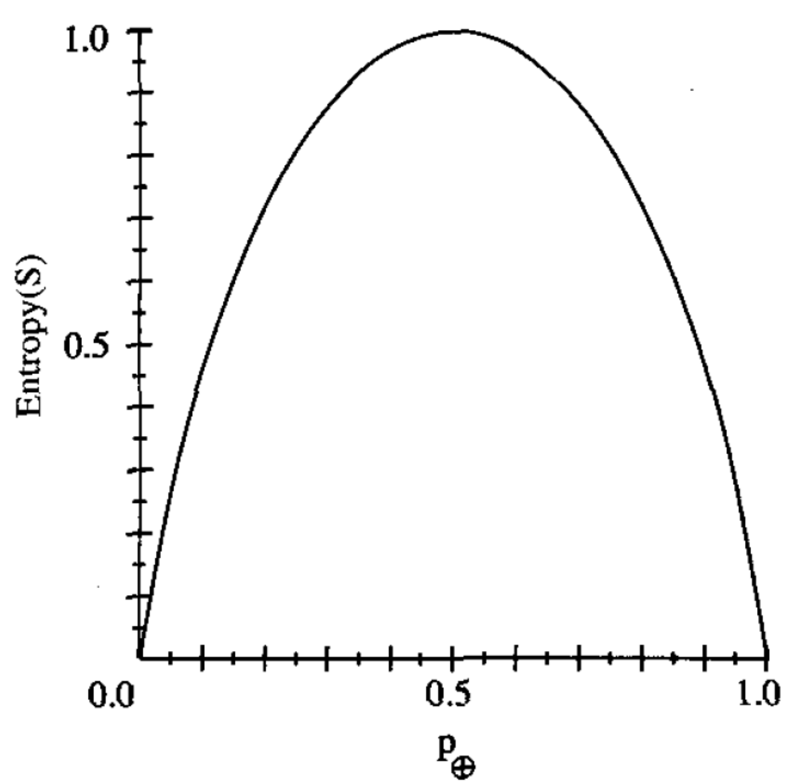
Entropy: Statistical measure to characterize the (im)purity of examples

$$
\begin{aligned}
\text{Entropy} &= -\, p_{No} \log_2 p_{No} - p_{Yes} \log_2 p_{Yes} \\
&= -(5/14) \log_2(5/14) - (9/14)\log_2(9/14) \\
&= 0.94
\end{aligned}
$$

# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

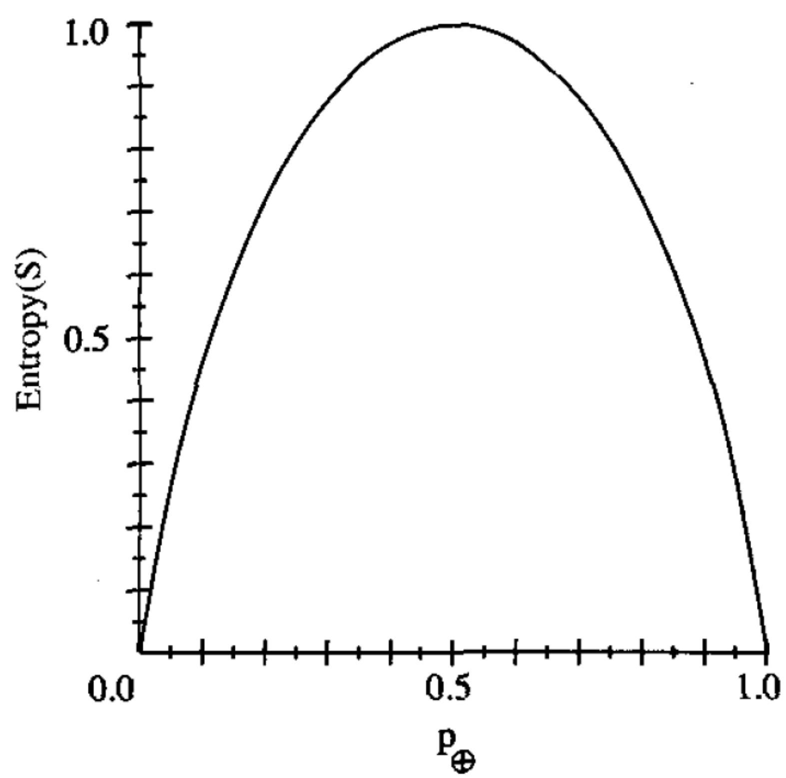
$$\begin{aligned} \text{Entropy} &= -\, p_{No} \log_2 p_{No} - p_{Yes} \log_2 p_{Yes} \\ &= -(5/14) \log_2(5/14) - (9/14)\log_2(9/14) \\ &= 0.94 \end{aligned}$$



# Entropy

Entropy: Statistical measure to characterize the (im)purity of examples

$$\begin{aligned}\text{Entropy} &= -\,p_{No}\log_2 p_{No} - p_{Yes}\log_2 p_{Yes}\\ &= -(5/14)\log_2(5/14) - (9/14)\log_2(9/14)\\ &= 0.94\end{aligned}$$




Avg. # of bits to transmit

# Information Gain

Information Gain: Reduction in entropy

# Information Gain

Information Gain: Reduction in entropy

By partitioning examples (S) on attribute A

# Information Gain

By partitioning examples (S) on attribute A

# Information Gain

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong
- S = [9+, 5-]

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong
- S = [9+, 5-]
- $S_{Weak}$ = [6+, 2-]

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong
- S = [9+, 5-]
- $S_{Weak}$ = [6+, 2-]
- $S_{Strong}$ = [3+, 3-]

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong
- S = [9+, 5-]
- $S_{Weak}$ = [6+, 2-]
- $S_{Strong}$ = [3+, 3-]
- Gain (S, Wind) = Entropy (S) - (8/14)*Entropy ($S_{Weak}$) - (6/14)*Entropy($S_{Strong}$)

# Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

| Wind | PlayTennis |
|---|---|
| Weak | No |
| Strong | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Strong | No |
| Strong | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Strong | Yes |
| Strong | Yes |
| Weak | Yes |
| Strong | No |

- A = Wind
- Values (Wind) = Weak, Strong
- S = [9+, 5-]
- $S_{Weak}$ = [6+, 2-]
- $S_{Strong}$ = [3+, 3-]
- Gain (S, Wind) = Entropy (S) - (8/14)*Entropy ($S_{Weak}$) - (6/14)*Entropy($S_{Strong}$)

= 0.048

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]
- $S_{Overcast}$ = [4+, 0-]

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]
- $S_{Overcast}$ = [4+, 0-]
- $S_{Rain}$ = [3+, 2-]

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]
- $S_{Overcast}$ = [4+, 0-]
- $S_{Rain}$ = [3+, 2-]
- Gain (S, Outlook) = Entropy (S) - (5/14)*Entropy ($S_{Sunny}$) - (4/14)*Entropy($S_{Overcast}$)-

# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]
- $S_{Overcast}$ = [4+, 0-]
- $S_{Rain}$ = [3+, 2-]
- Gain (S, Outlook) = Entropy (S) - (5/14)*Entropy ($S_{Sunny}$) - (4/14)*Entropy($S_{Overcast}$)- (5/14)*Entropy($S_{Rain}$)

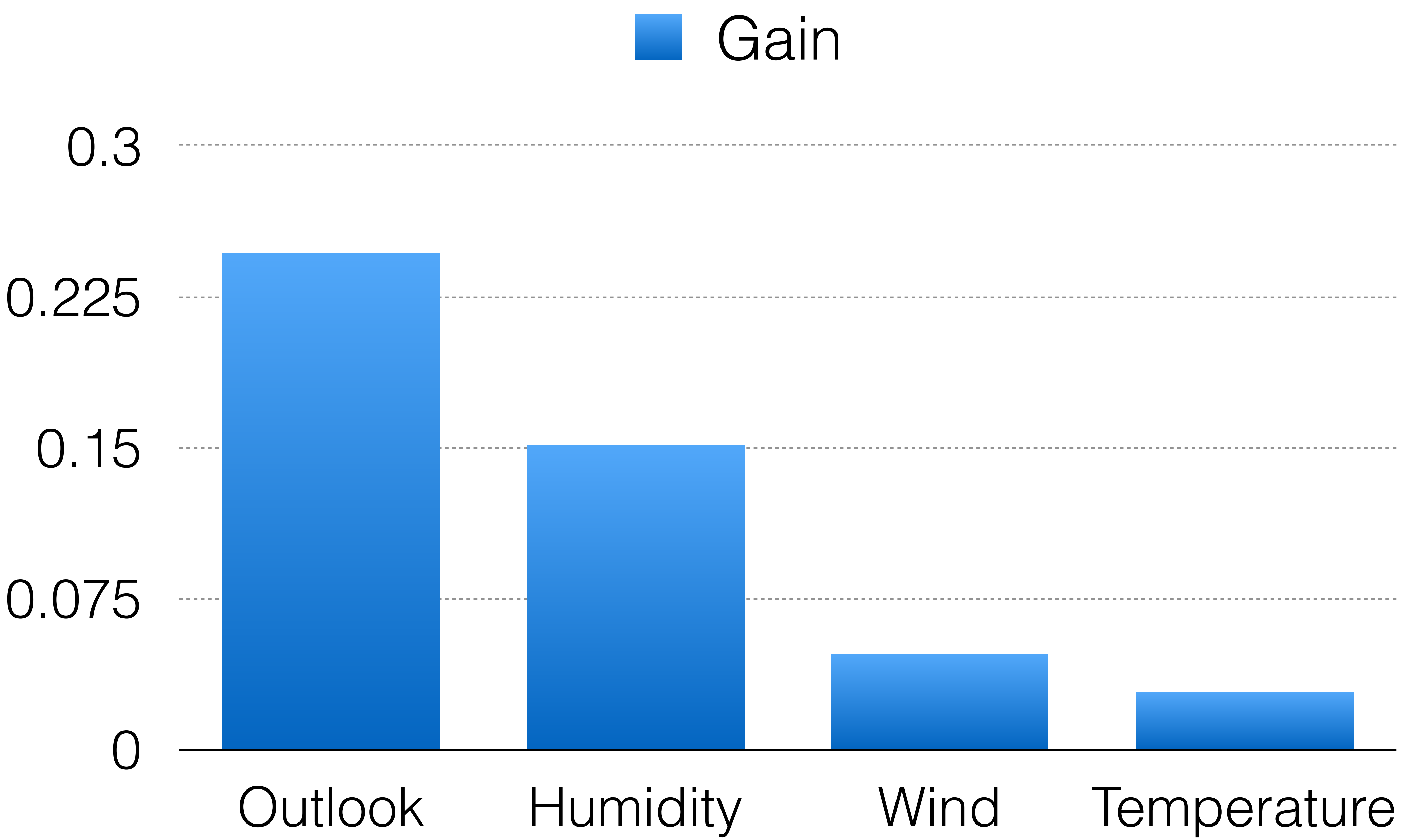
# Information Gain

| Outlook | PlayTennis |
|---|---|
| Sunny | No |
| Sunny | No |
| Overcast | Yes |
| Rain | Yes |
| Rain | Yes |
| Rain | No |
| Overcast | Yes |
| Sunny | No |
| Sunny | Yes |
| Rain | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Overcast | Yes |
| Rain | No |

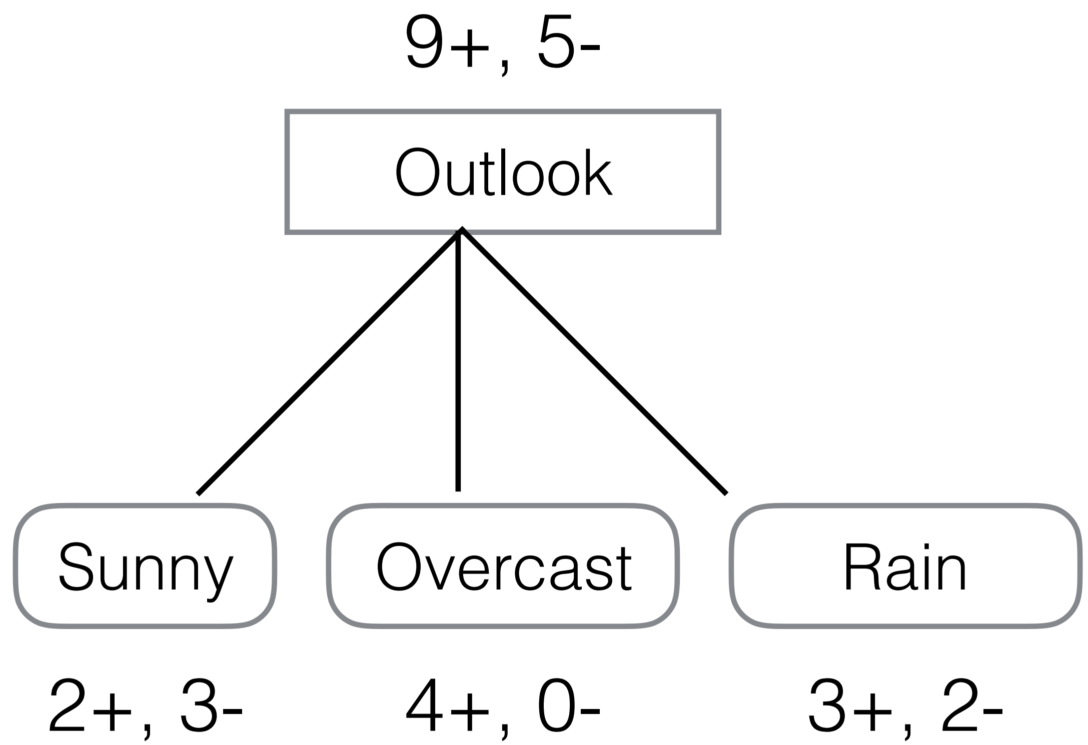
- A = Outlook
- Values (Outlook) = Sunny, Overcast, Rain
- S = [9+, 5-]
- $S_{Sunny}$ = [2+, 3-]
- $S_{Overcast}$ = [4+, 0-]
- $S_{Rain}$ = [3+, 2-]
- Gain (S, Outlook) = Entropy (S) - (5/14)*Entropy ($S_{Sunny}$) - (4/14)*Entropy($S_{Overcast}$)- (5/14)*Entropy($S_{Rain}$)

= 0.246

# Information Gain

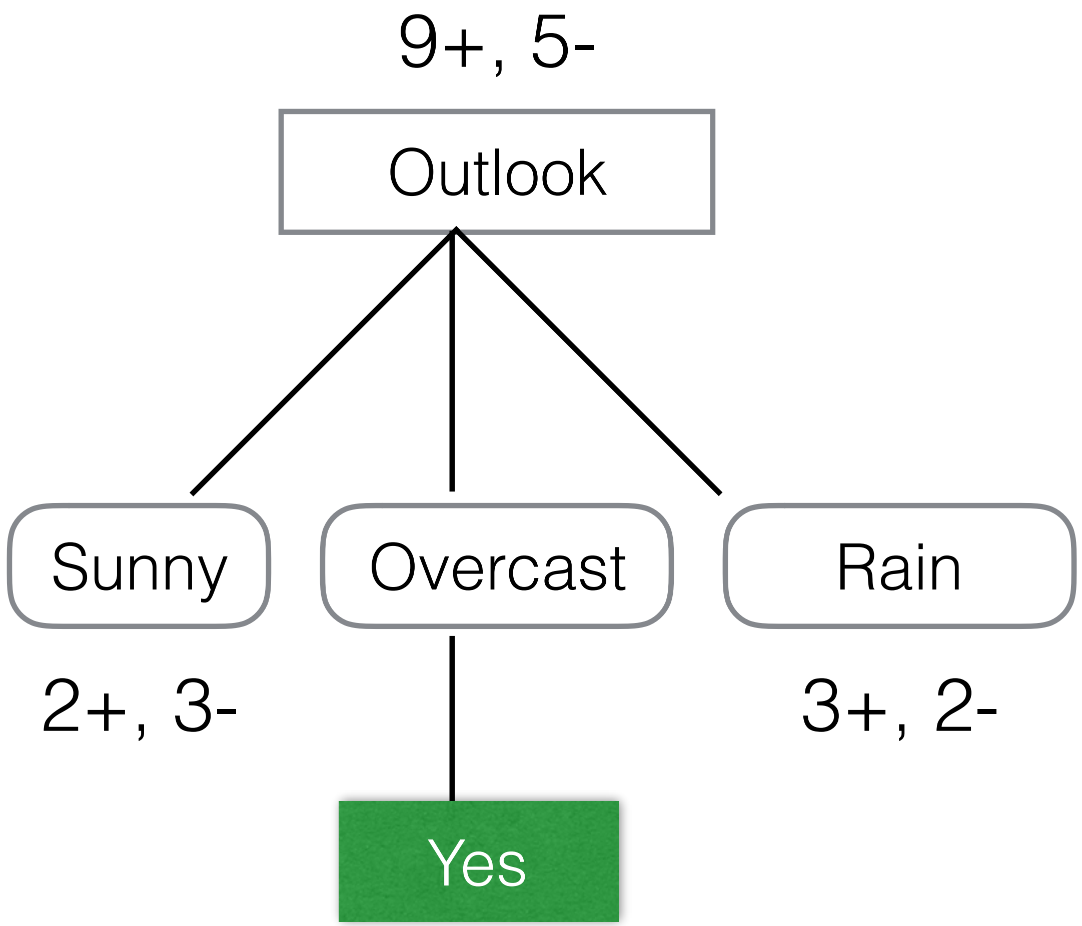


# Worked Out Example




Example borrowed from Tom Mitchell's text book

# Worked Out Example




Example borrowed from Tom Mitchell's text book

# Worked Out Example

9+, 5-

Outlook

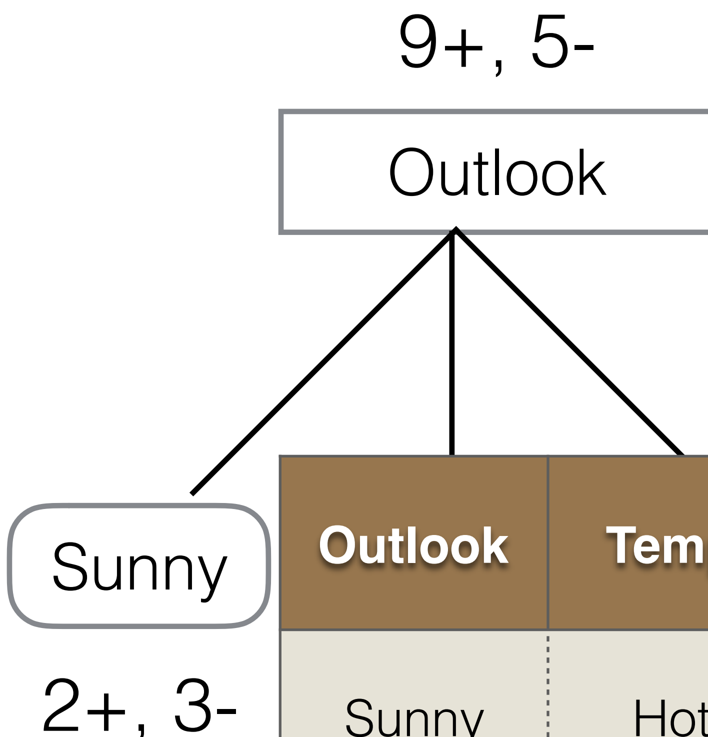
Sunny

2+, 3-

| Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Sunny | Mild | High | Strong | Yes |

Example borrowed from Tom Mitchell's text book

# Worked Out Example




S' = S_outlook is sunny

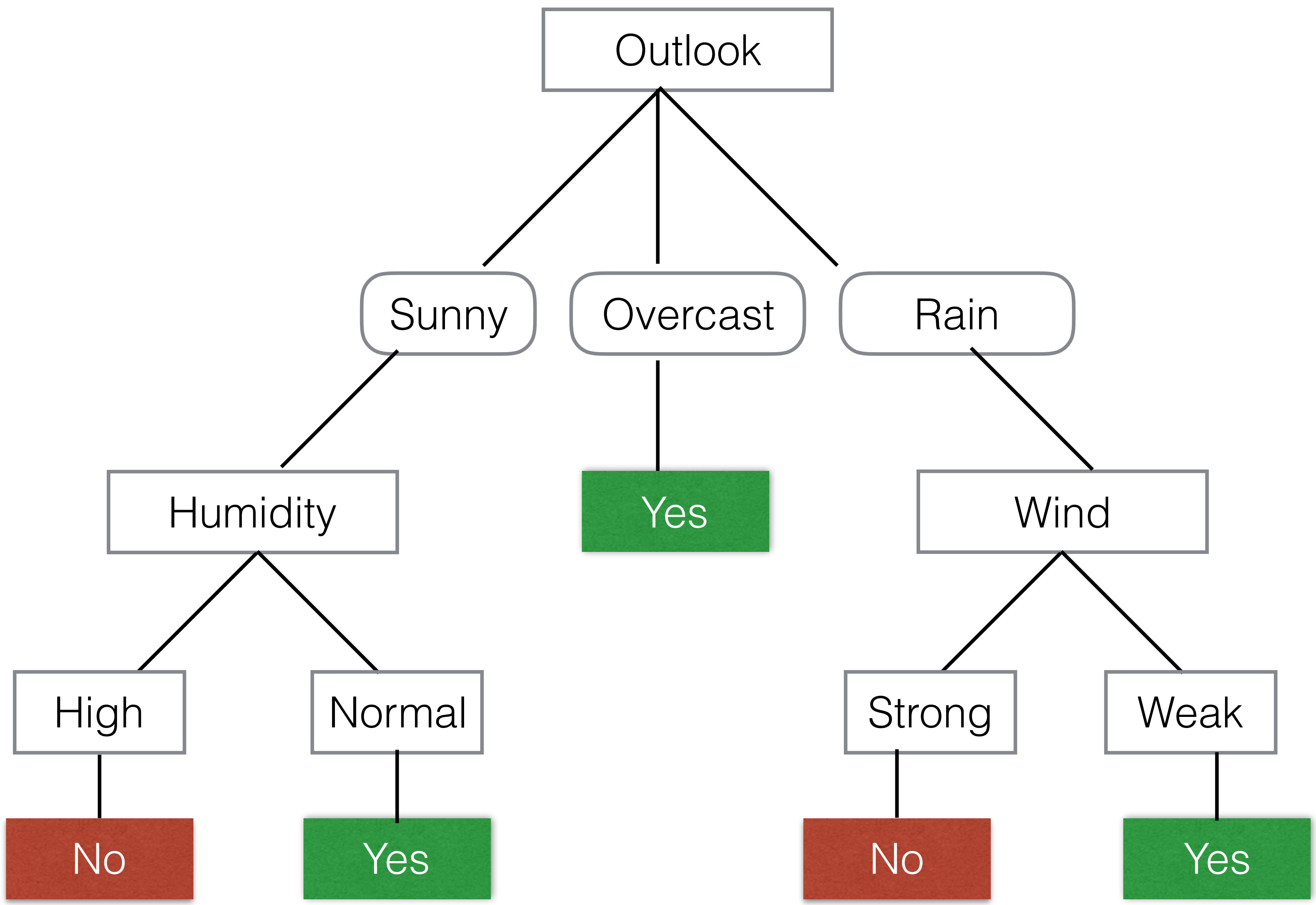
Entropy (S') =

$-(2/5) \log_2(2/5) - (3/5)\log_2(3/5)$

| Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Sunny | Mild | High | Strong | Yes |

Example borrowed from Tom Mitchell's text book

# Worked Out Example




Example borrowed from Tom Mitchell's text book