Artificial Intelligence

COMP3411

Week 2, Term 3, 2025

Maryam Hashemi (m.Hashemi@unsw.edu.au)

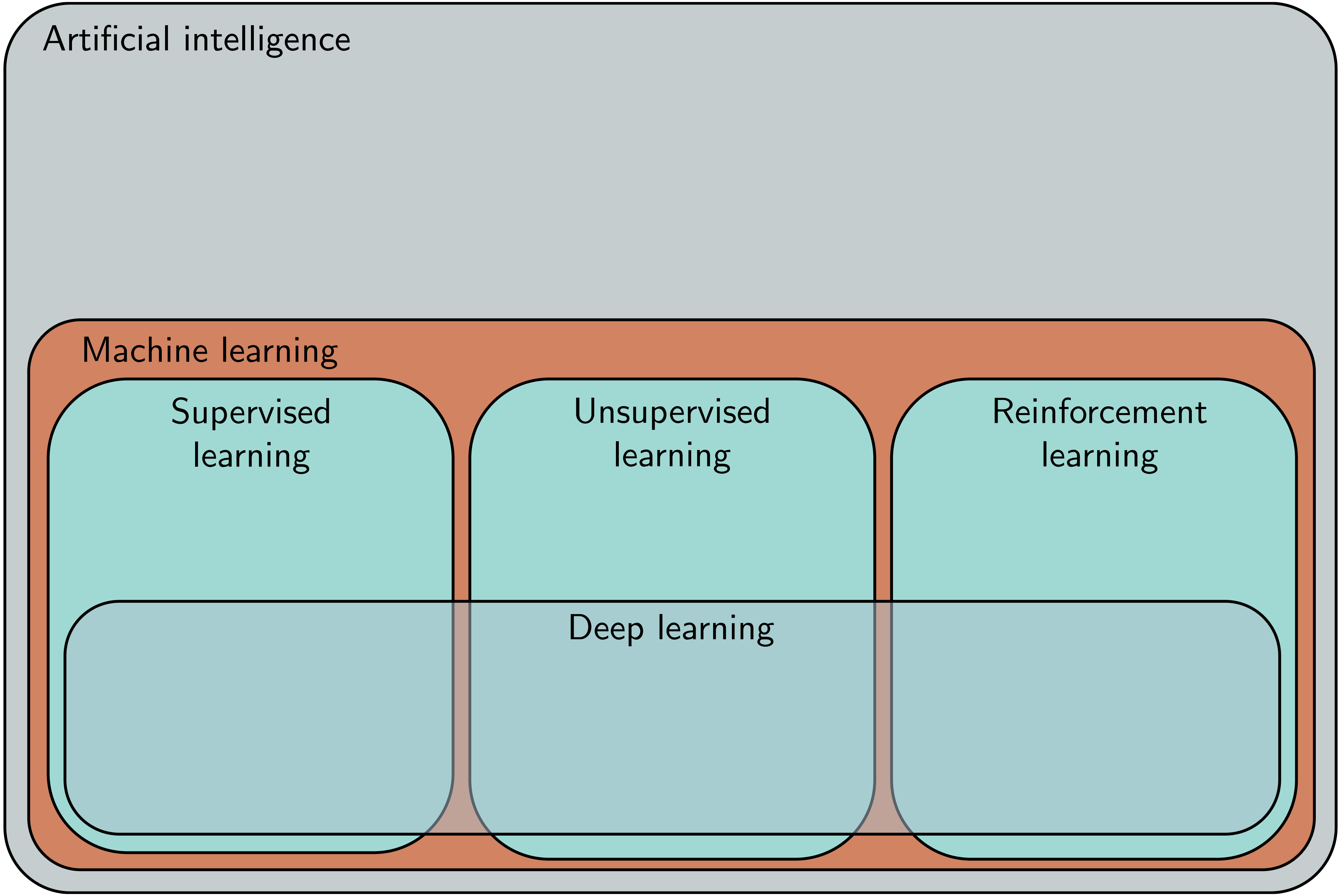
Overview

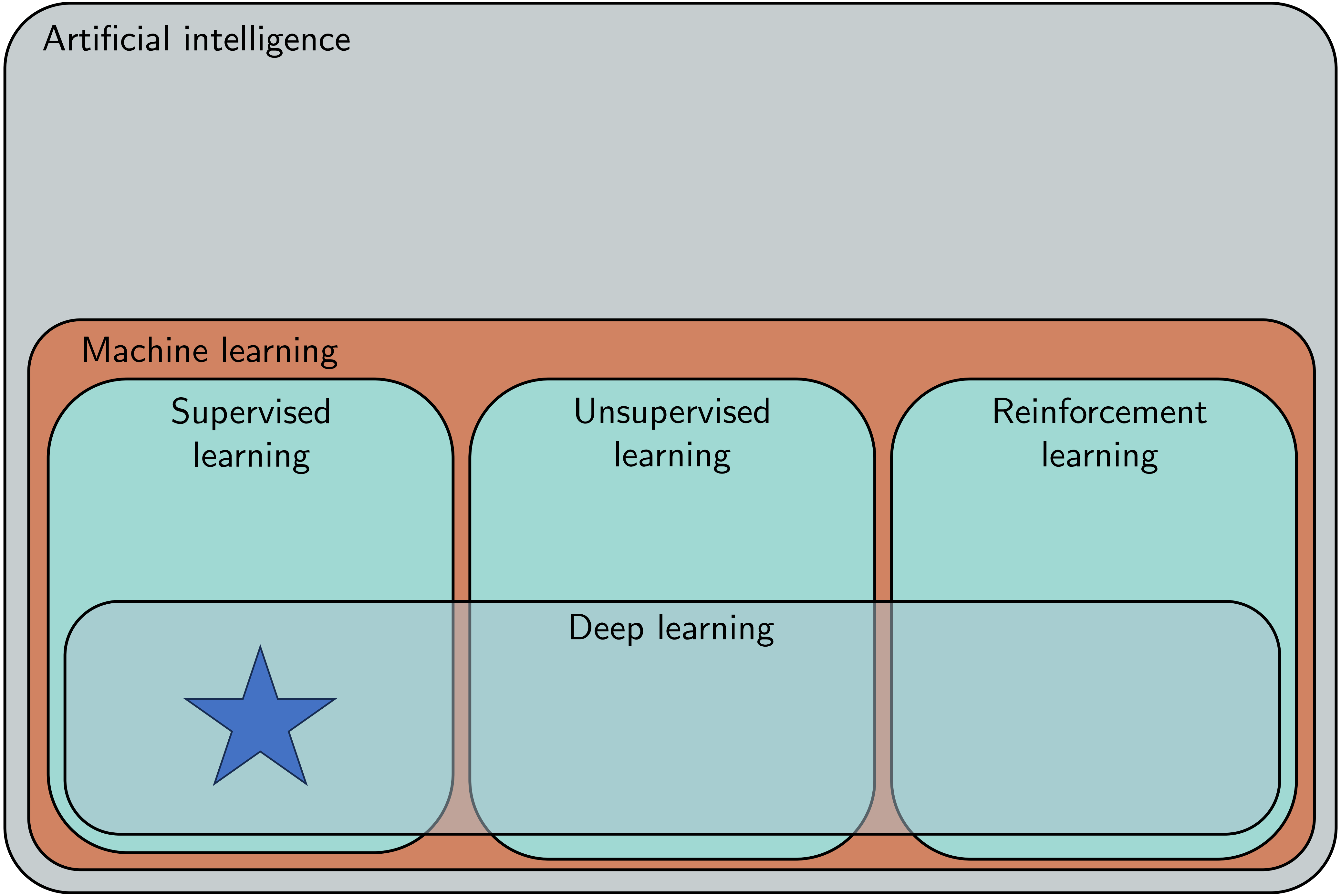
✓**Different Subfields of AI Algorithms.**

✓Supervised Learning: Decision Trees.

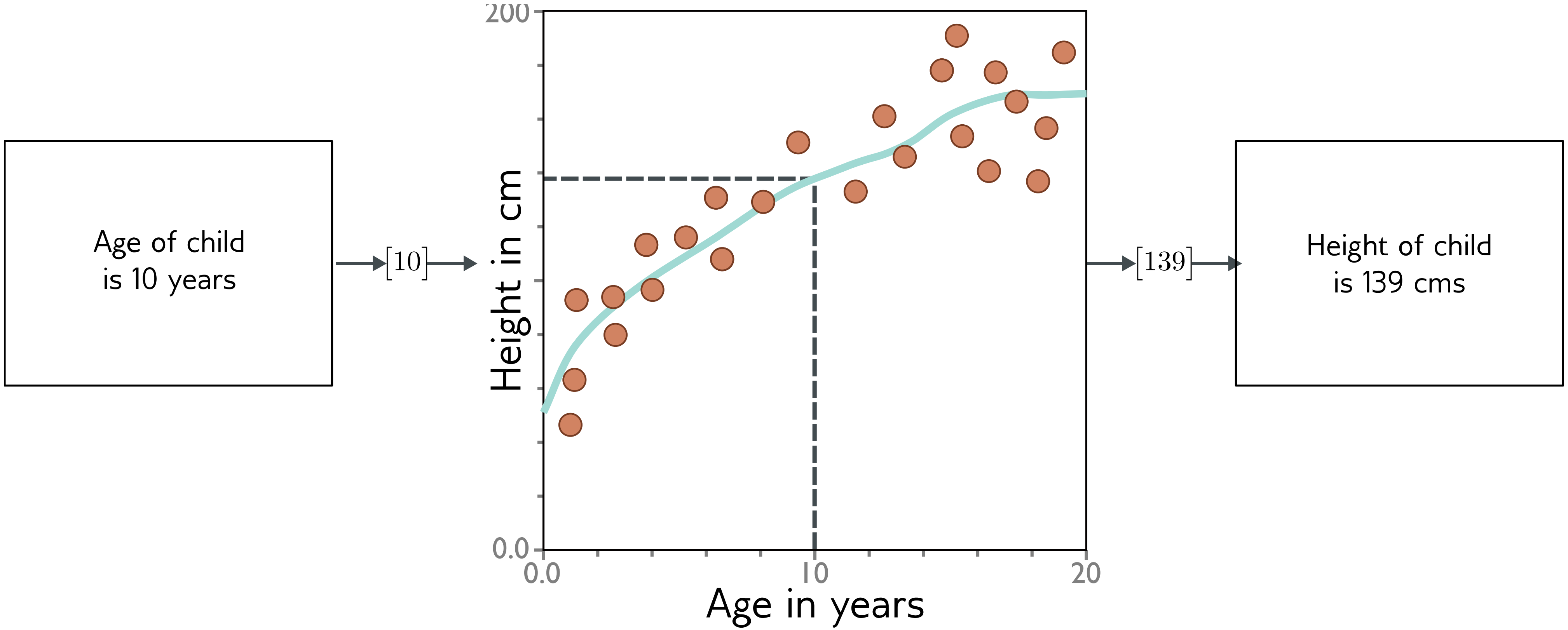
✓Ensemble Learning:

- Bagged Trees.
- Random Forests.
- Boosting Trees.


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

What Is a Supervised Learning Model?

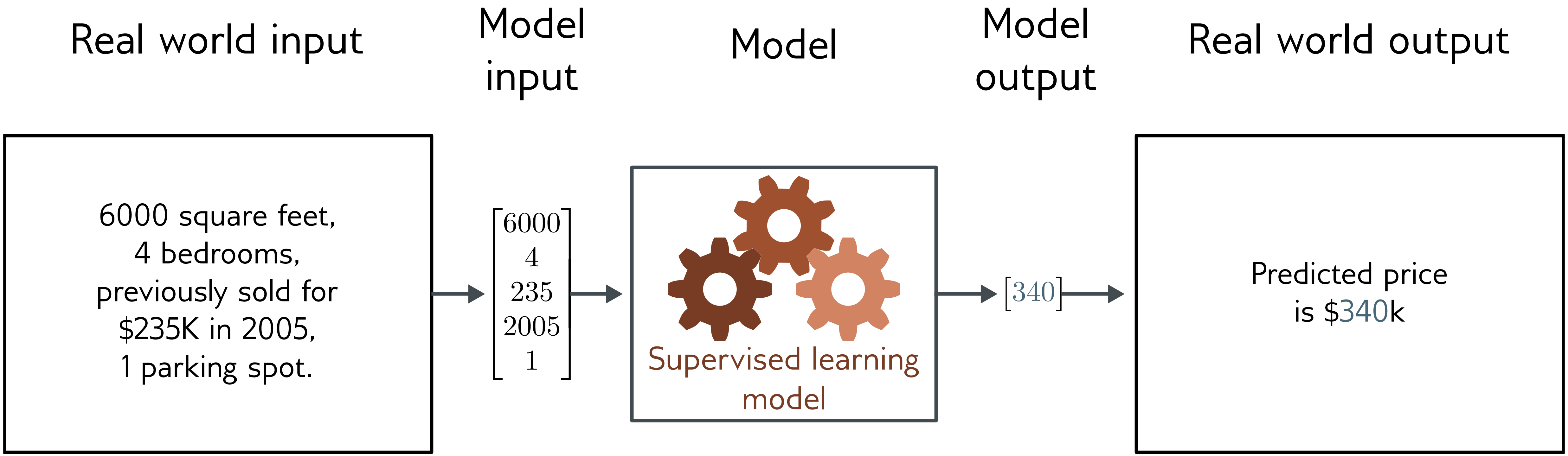


Age of child
is 10 years

Height of child
is 139 cms

- An equation relating input (age) to output (height)
- Search through family of possible equations to find one that fits training data well

Supervised Learning, Regression



Real world input

Model input

Model

Model output

Real world output

6000 square feet,
4 bedrooms,
previously sold for
$235K in 2005,
1 parking spot.

$$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$$

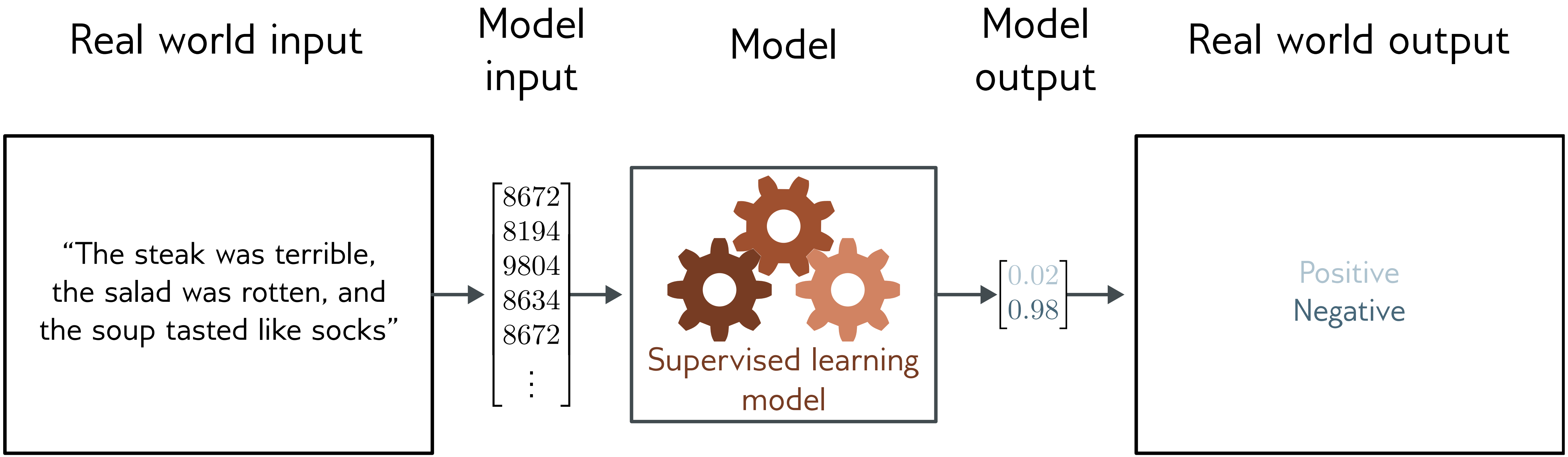
Supervised learning model

$$[340]$$

Predicted price
is $340k

- Univariate regression problem (one output, real value)

Supervised Learning, Binary Classification



Real world input

Model input

Model

Model output

Real world output

"The steak was terrible, the salad was rotten, and the soup tasted like socks"

$$\begin{bmatrix} 8672 \\ 8194 \\ 9804 \\ 8634 \\ 8672 \\ \vdots \end{bmatrix}$$

Supervised learning model

$$\begin{bmatrix} 0.02 \\ 0.98 \end{bmatrix}$$

Positive

Negative

- Binary classification problem (two discrete classes)

Supervised Learning, Multiclass Classification

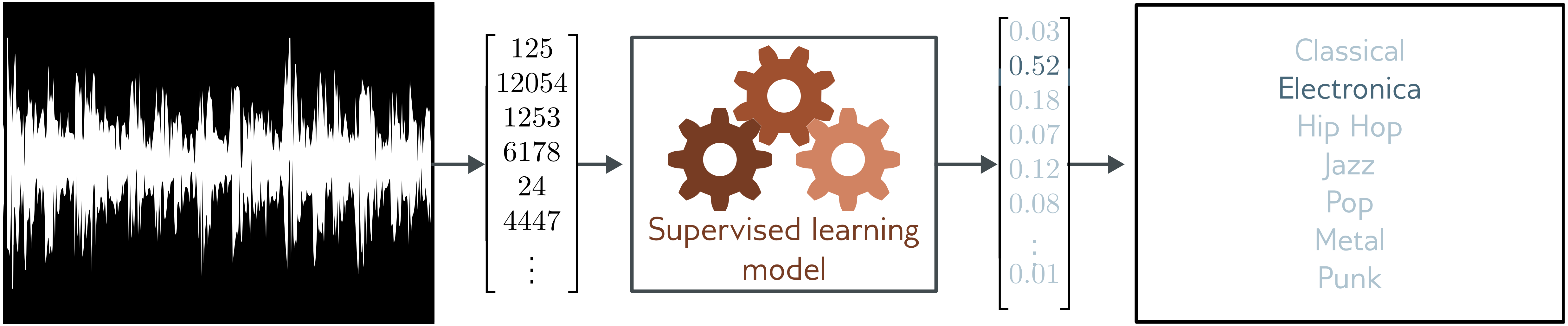
Real world input

Model input

Model

Model output

Real world output



$$\begin{bmatrix} 125 \\ 12054 \\ 1253 \\ 6178 \\ 24 \\ 4447 \\ \vdots \end{bmatrix}$$

Supervised learning model

$$\begin{bmatrix} 0.03 \\ 0.52 \\ 0.18 \\ 0.07 \\ 0.12 \\ 0.08 \\ \vdots \\ 0.01 \end{bmatrix}$$

Classical
Electronica
Hip Hop
Jazz
Pop
Metal
Punk

- Multiclass classification problem (discrete classes, >2 possible values)

Supervised Learning, Multiclass Classification

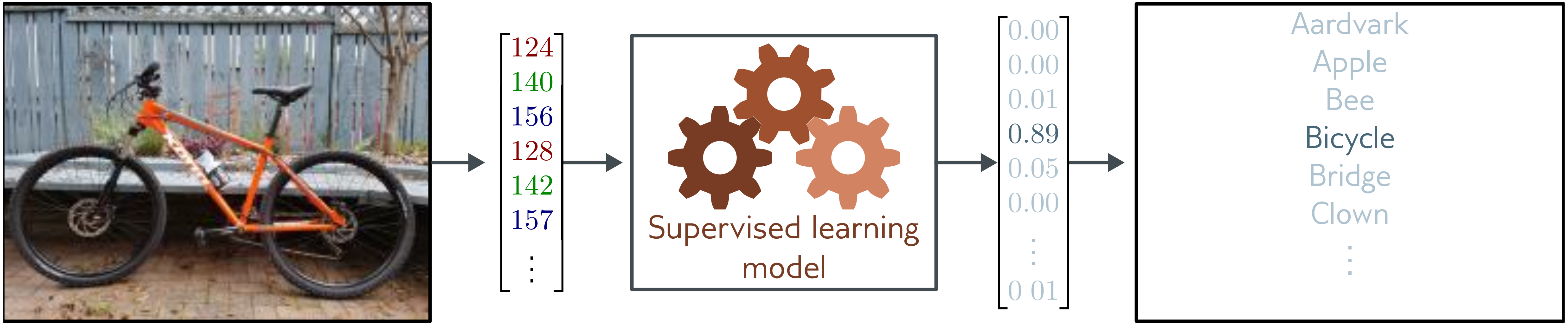
Real world input

Model input

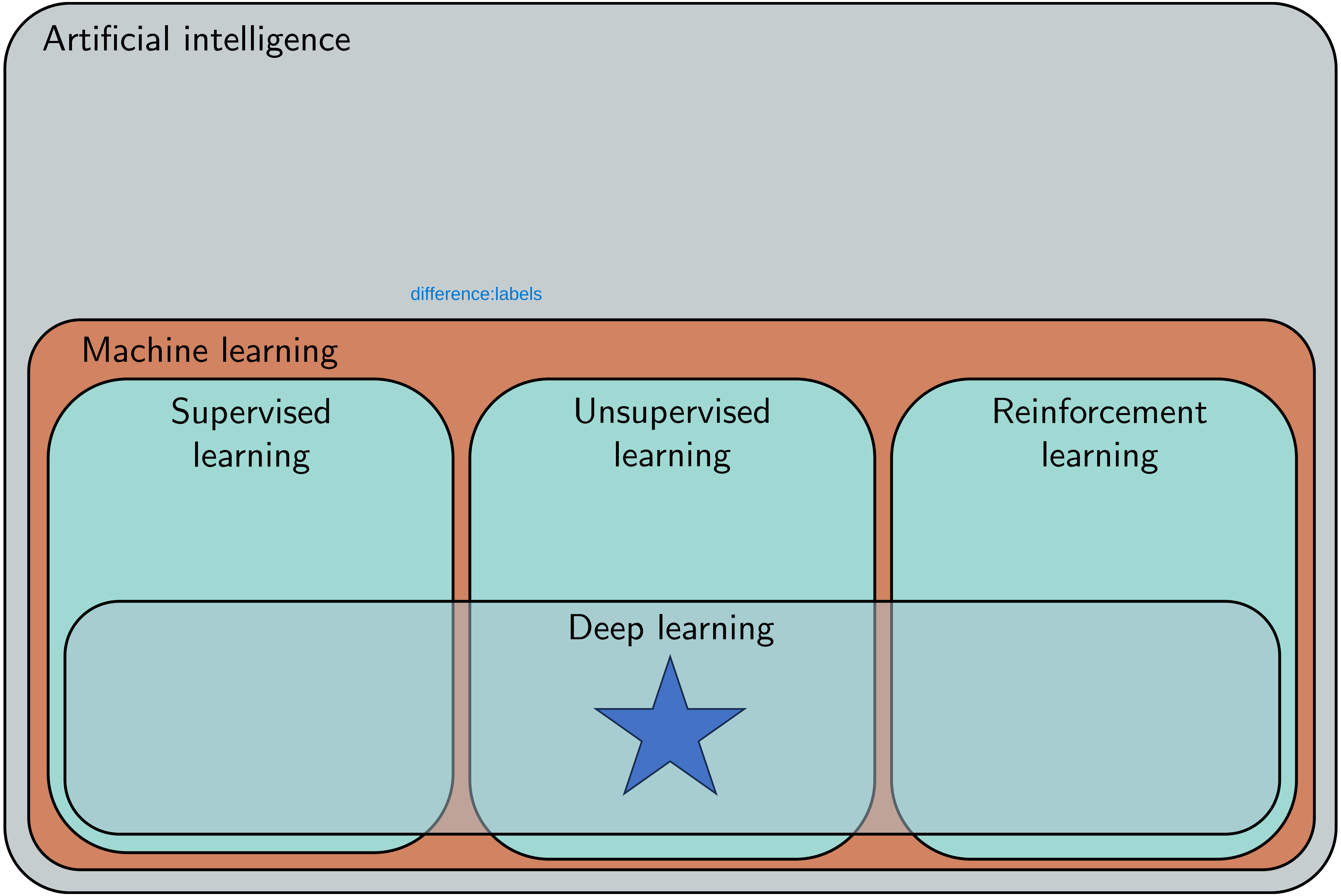
Model

Model output

Real world output



- Multiclass classification problem (discrete classes, >2 possible classes)



Artificial intelligence

difference:labels

Machine learning

Supervised learning

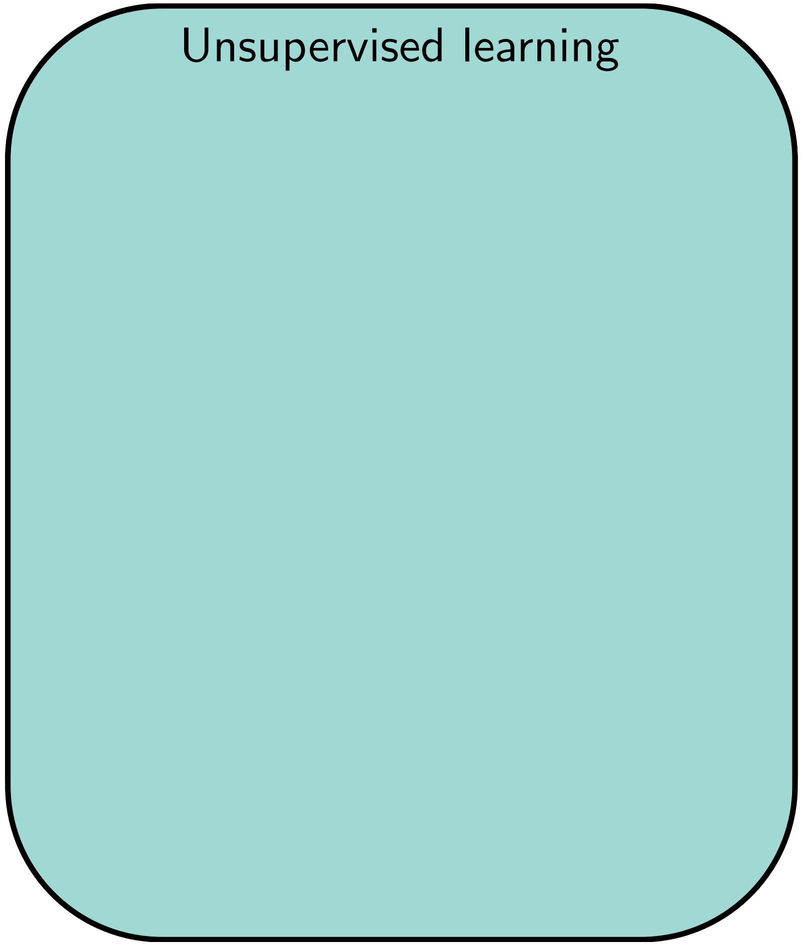
Unsupervised learning

Reinforcement learning

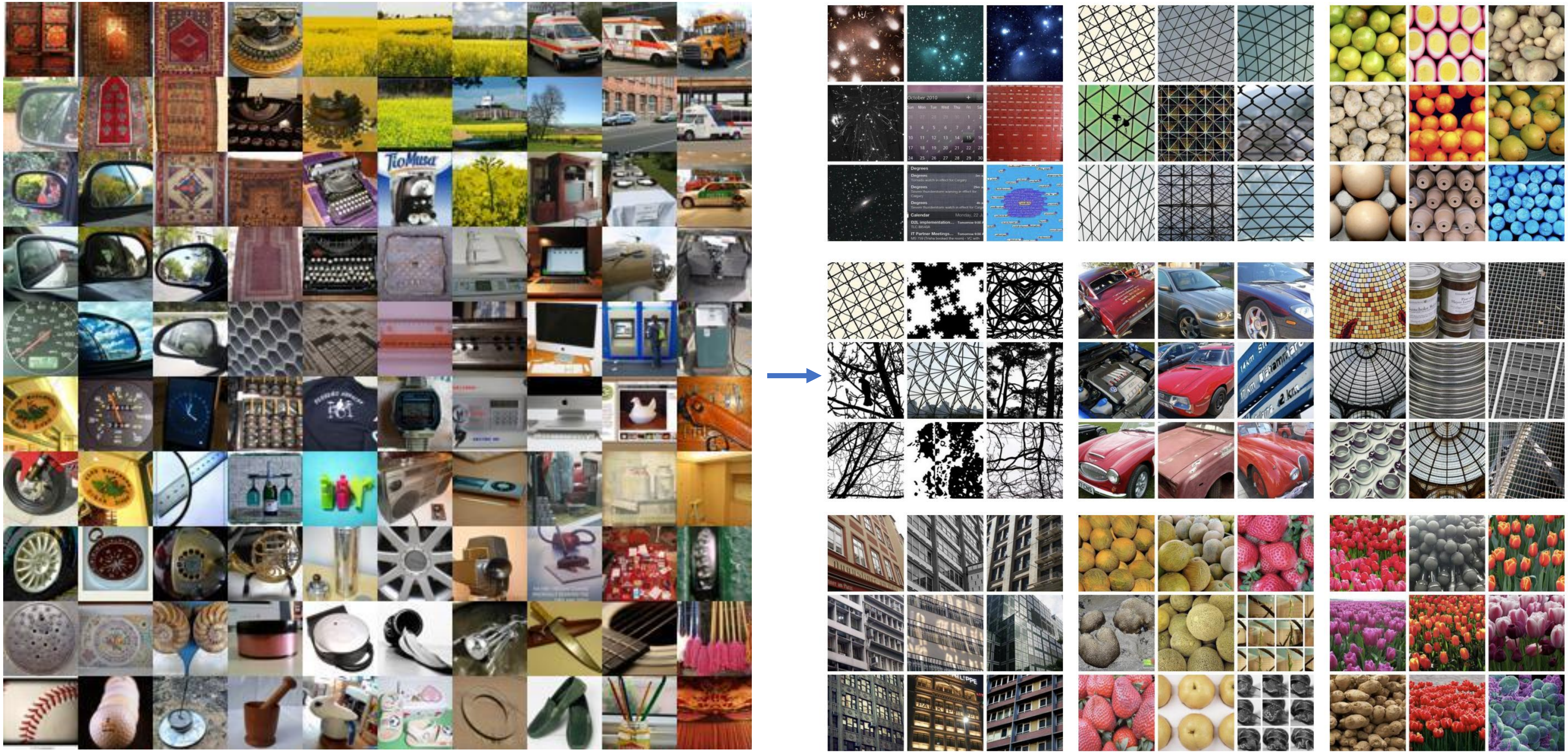
Deep learning

Unsupervised Learning

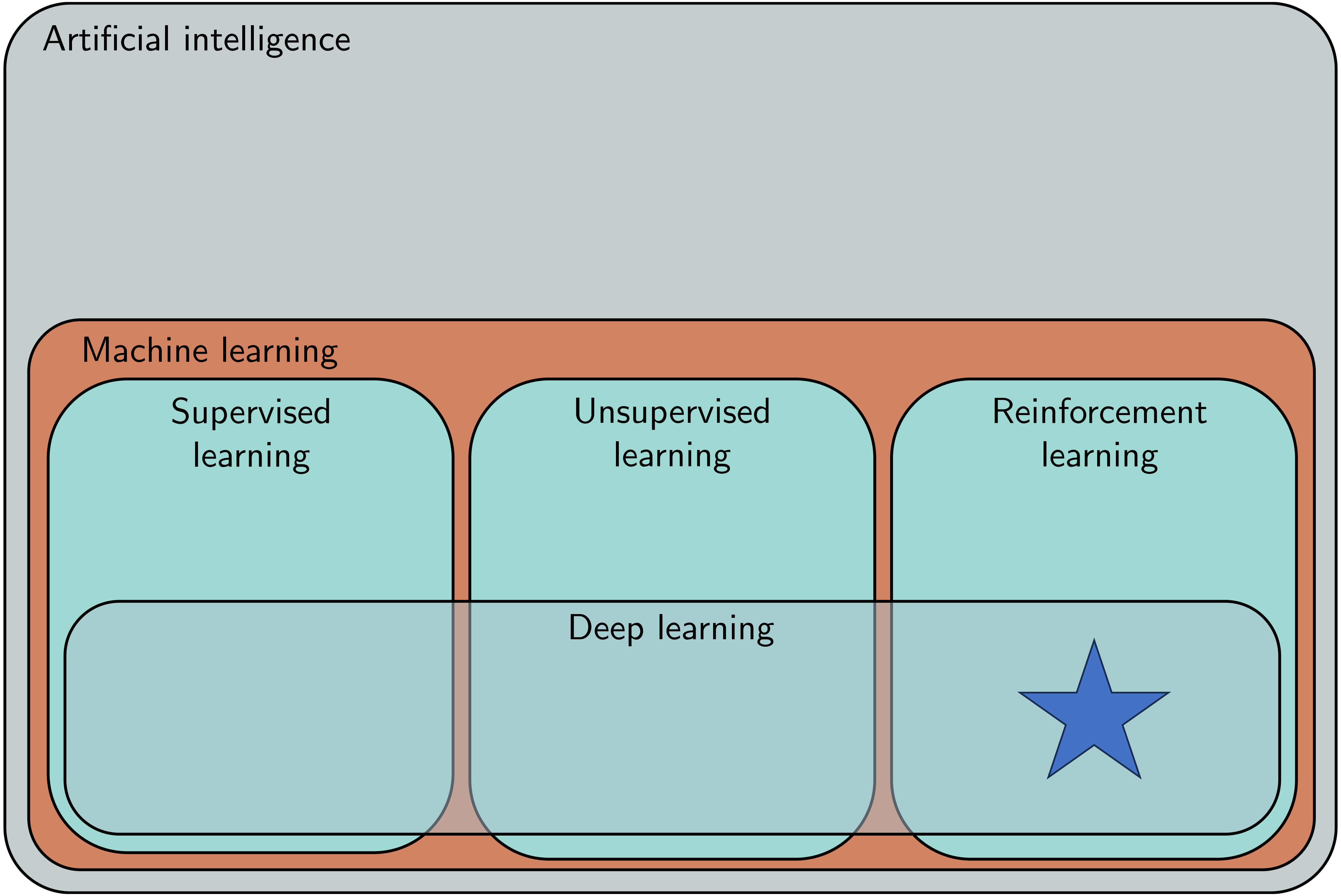
- Learning about a dataset without labels
  - Clustering: Grouping similar data points together.



Unsupervised learning



DeepCluster: Deep Clustering for Unsupervised Learning of Visual Features (Caron et al., 2018)


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Reinforcement Learning

- A set of states
- A set of actions
- A set of rewards

- Goal: take actions to change the state so that you receive rewards

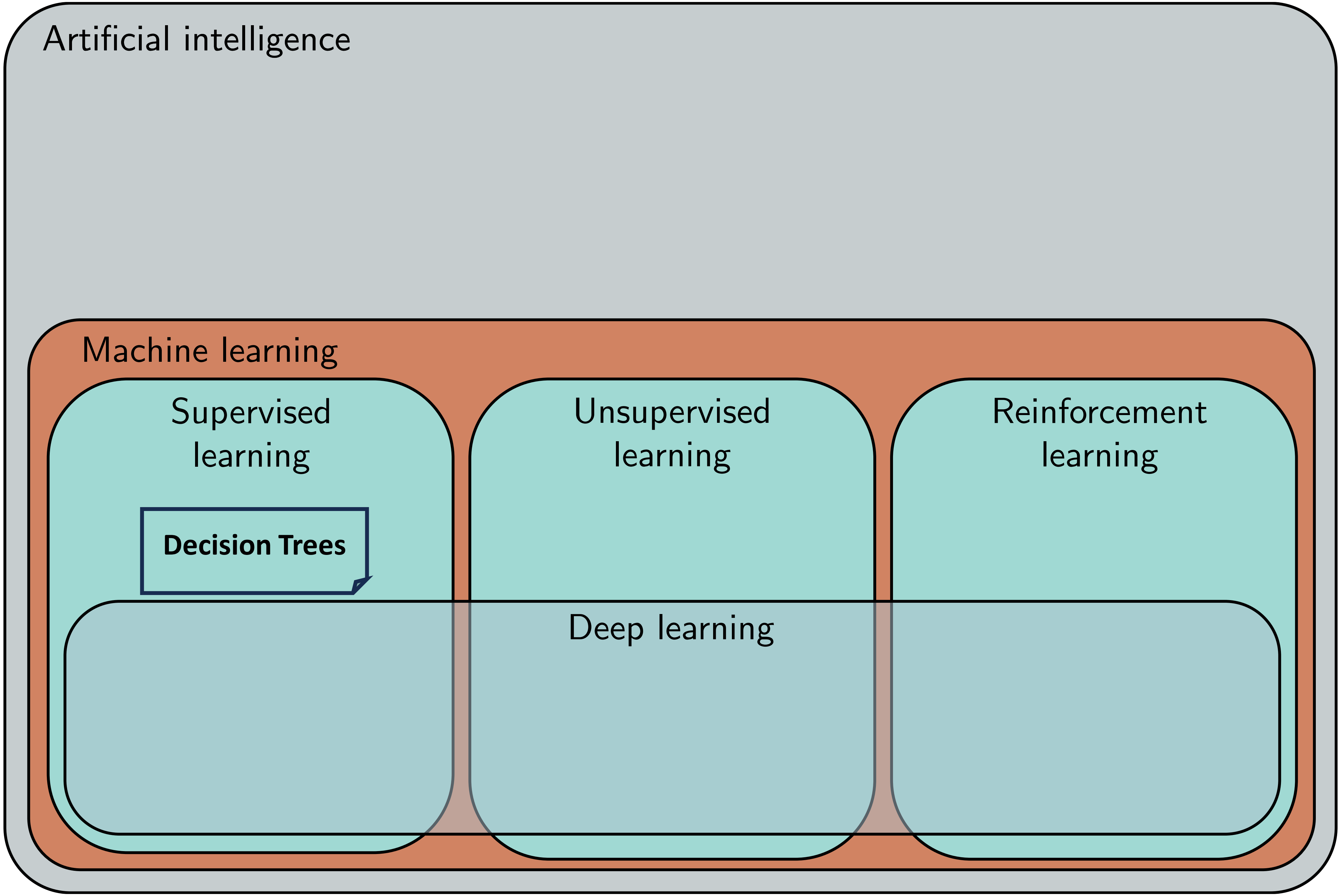
- You have to explore the environment yourself to gather data as you go. Very much like an infant learning.

✓Different Subfields of AI Algorithms.

✓**Supervised Learning: Decision Trees.**

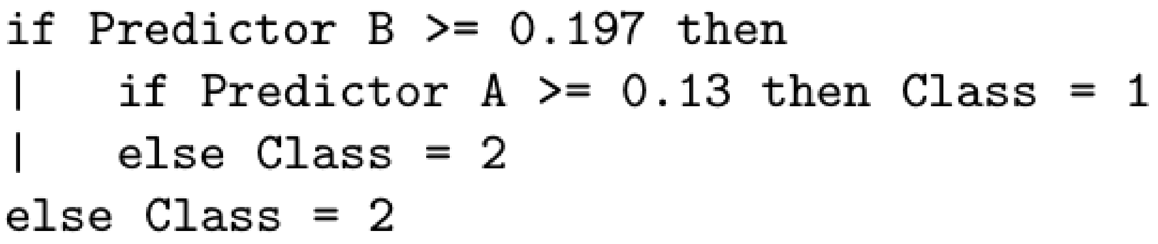
✓Ensemble Learning:

- Bagged Trees.
- Random Forests.
- Boosting Trees.


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Decision Trees
Deep learning

Decision Trees

- Decision trees are a classical **supervised machine learning** method.
- They consist of a sequence of nested *if–then* rules applied to the predictor variables (features), which recursively partition the data space.



```
if Predictor B >= 0.197 then
|    if Predictor A >= 0.13 then Class = 1
|    else Class = 2
else Class = 2
```

- Because the splitting rules can be naturally represented in a tree structure, these models are referred to as decision-tree methods.
- Decision trees are widely used for both *classification* and *regression* tasks.

Decision Trees

- Basic decision trees partition the data into subsets that are increasingly **homogeneous** with respect to the response variable.

if Predictor A >= 0.13 and Predictor B >= 0.197 then Class = 1 if
Predictor A >= 0.13 and Predictor B < 0.197 then Class = 2 if
Predictor A < 0.13 then Class = 2

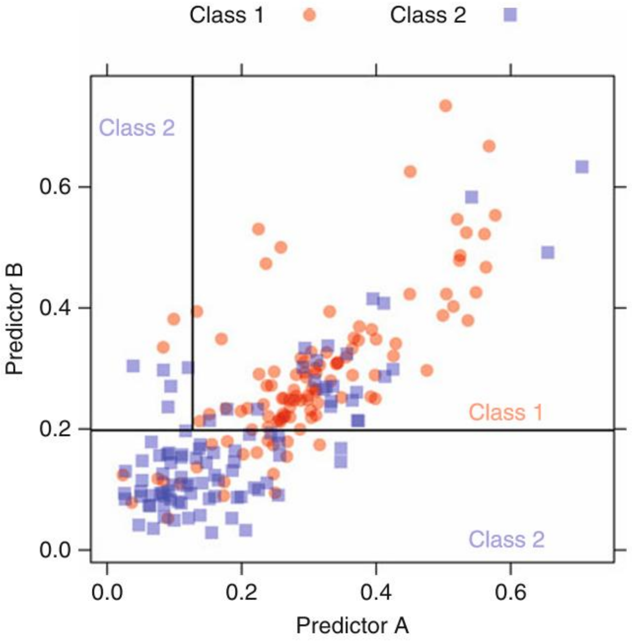
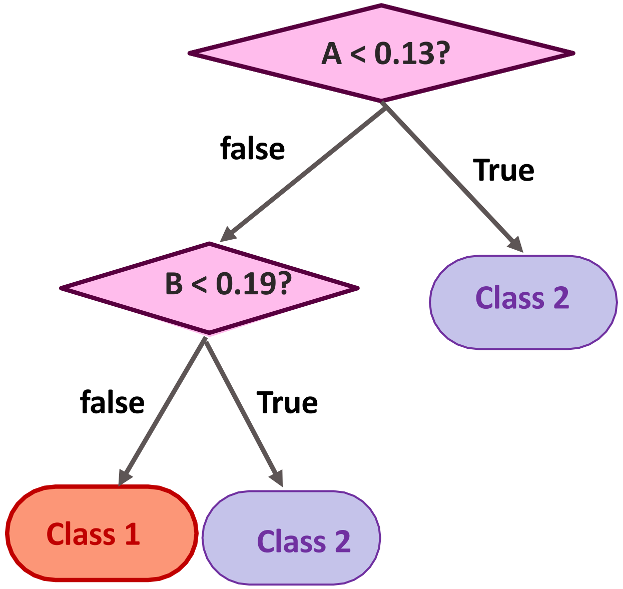


Fig. 1: An example of the predicted classes within regions defined by a tree-based model.



A < 0.13?

false

True

B < 0.19?

Class 2

false

True

Class 1

Class 2

Decision Trees

- Decision trees are typically drawn upside down, with the root node at the top and the leaves at the bottom.
- For example, Feature A may be the most important feature in determining the classification, so we want to check it earlier.
- The ultimate goal is to construct a decision tree that *generalizes* well from the training data and **accurately classifies previously unseen samples**.
- However, several key questions arise:
  - How do we construct such a tree?
  - How do we determine the most informative feature (predictor) to split on first?

Decision Trees, Entropy

- Entropy is a measure of **randomness or uncertainty** in a random variable.
  - Higher entropy means more randomness
  - "Information" (about distribution) reduces entropy
  - It is maximized when all outcomes are equally likely.
  - It is minimized when the probability distribution is highly concentrated around a single outcome.

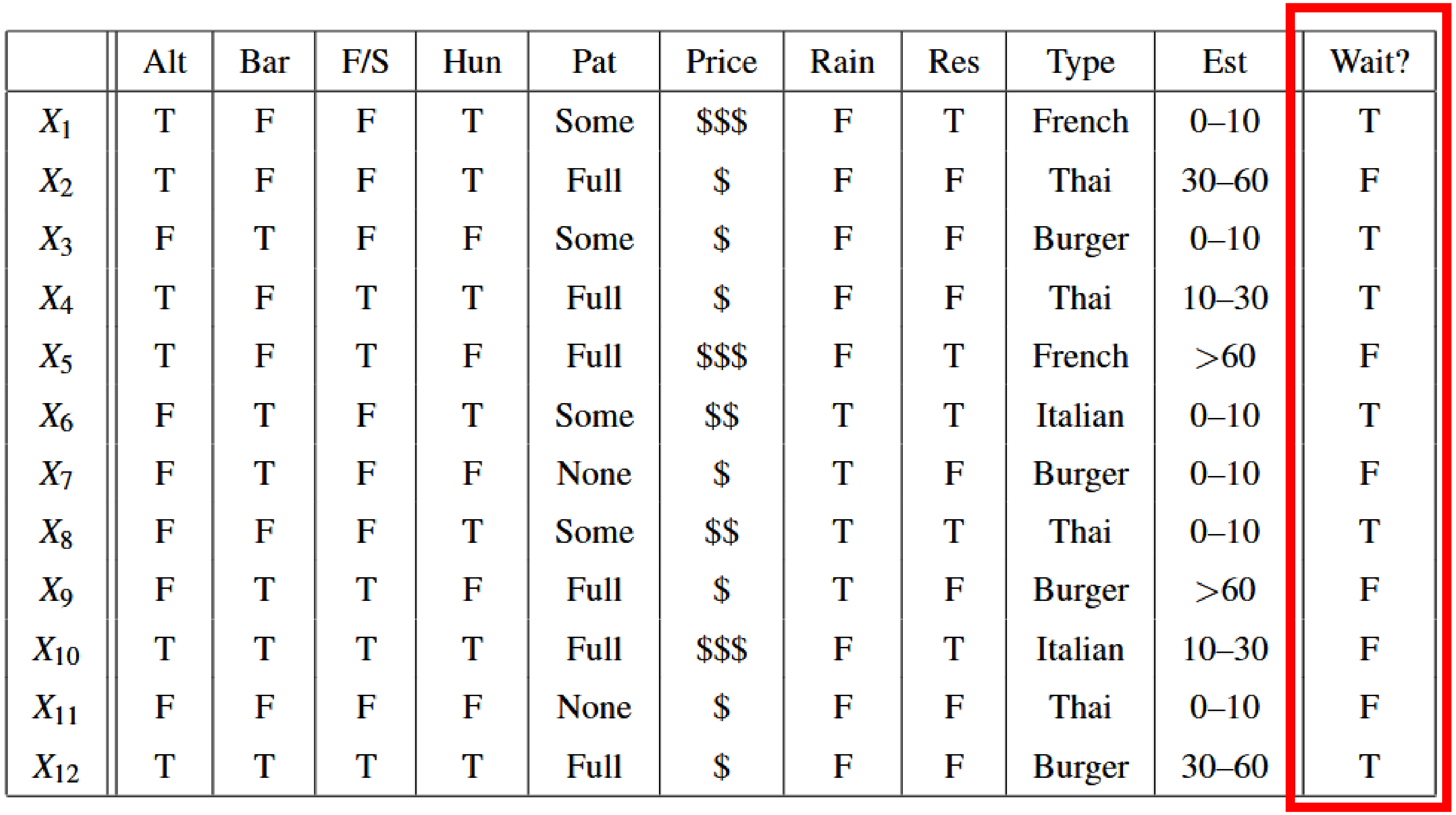
- Idea: Split based on information gain and measure information gain.

Definition: If the prior probabilities of *n* attribute values are $p_1, \cdots, p_n$, then the entropy of the distribution is:

$$H(\langle \mathrm{p}_1, \cdots, \mathrm{pn} \rangle) = \sum_{i=1}^{n} -\mathrm{p}_i \log_2 p_i$$

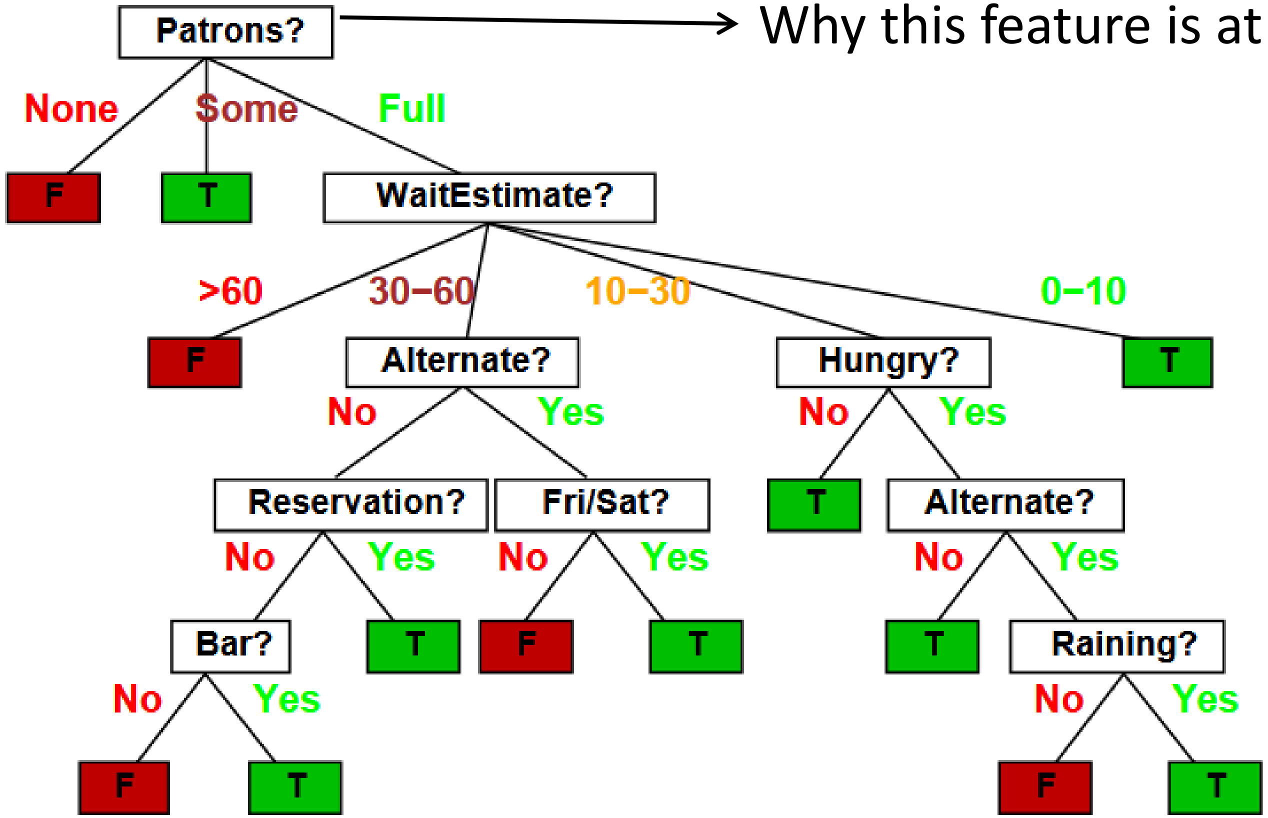
Decision Trees, Example

Suppose we want to train a decision tree to predict whether a person should wait in a restaurant queue. The decision is based on several features, such as whether an alternative restaurant is available, whether it is Friday or Saturday, the number of patrons currently present, whether the restaurant accepts reservations, and the estimated waiting time.



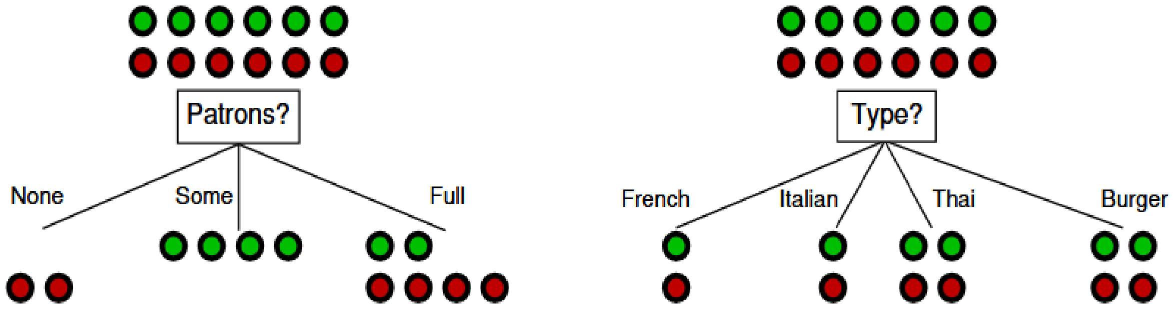
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Wait?
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	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

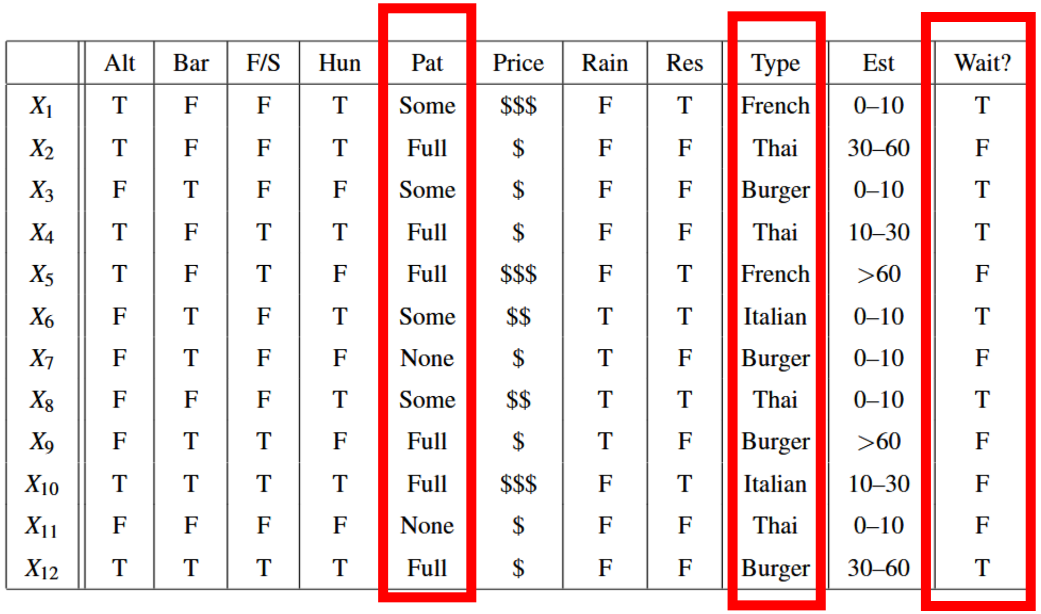
Decision Trees, Example



→ Why this feature is at the root of the tree?

Decision Trees, Example



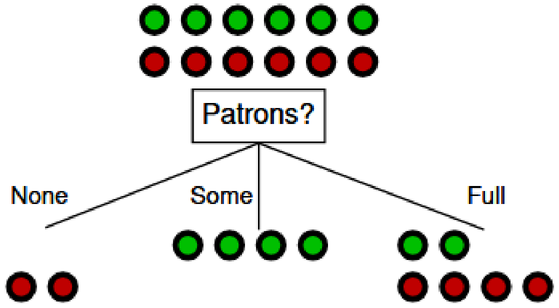


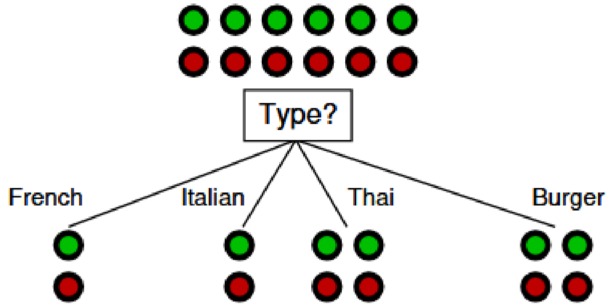
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Wait?
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	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

- Feature Patrons is considered more informative than Type, because it divides the training examples into subsets that are closer to being entirely positive or entirely negative (i.e., **more homogeneous**).
- This notion of “informativeness” can be formally quantified using the concept of **entropy**.
- A decision tree can therefore be constructed by selecting, at each step, the feature that minimizes entropy (or equivalently maximizes information gain).

Decision Trees, Example

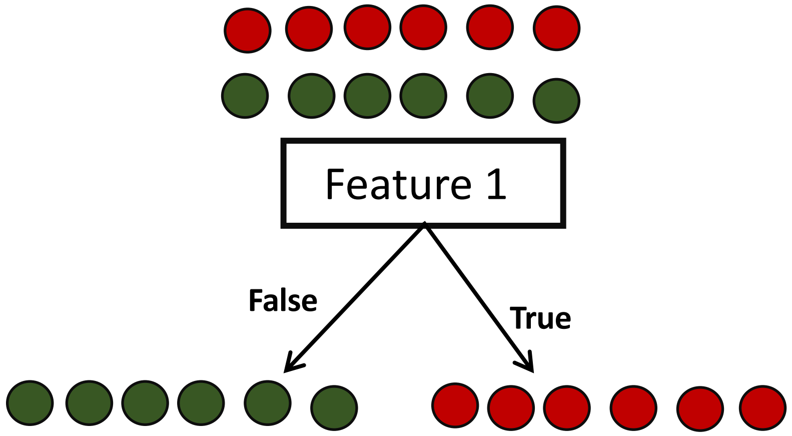
$$H(\langle \mathrm{p}_1,\cdots,\mathrm{pn}\rangle) \;=\; \sum_{i=1}^{n} -\mathrm{p}_i \log_2 p_i$$



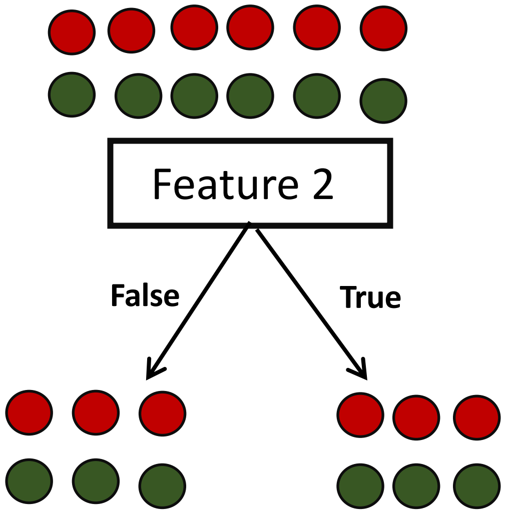


$$\begin{aligned}
\text{For Patrons, Entropy} &= \frac{1}{6}(0)+\frac{1}{3}(0)+\frac{1}{2}[-\frac{1}{3}\log_2(\frac{1}{3})-\frac{2}{3}\log_2(\frac{2}{3})] \\
&= 0+0+\frac{1}{2}[\frac{1}{3}(1.585)+\frac{2}{3}(0.585)] = 0.459 \\
\text{For Type, Entropy} &= \frac{1}{6}(1)+\frac{1}{6}(1)+\frac{1}{3}(1)+\frac{1}{3}(1) = 1
\end{aligned}$$

Decision Trees, Example

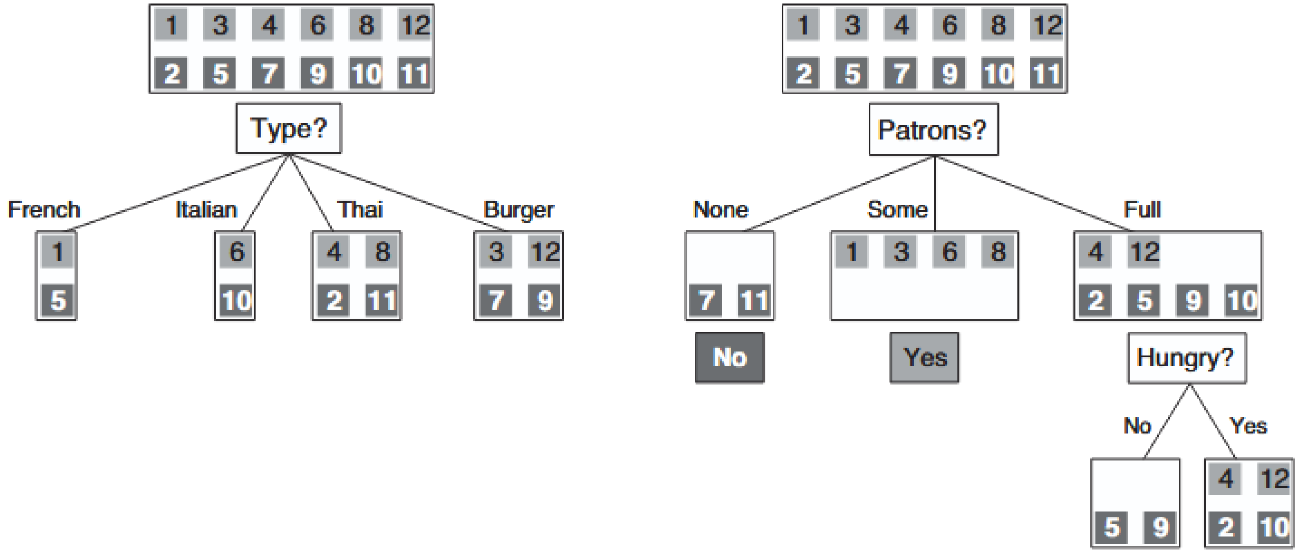


- At depth *d* of the tree, feature 1 is the **best** feature to evaluate.
- Feature 1 minimizes entropy (uncertainty) and separates the data into more homogeneous subsets.
- This conclusion is reached by calculating the entropy (**E**$\mathbf{E_{feature1} < E_{feature2}}$**).**



- At depth *d* of the tree, feature 2 is the **worst** feature to evaluate.
- Feature 2 maximize entropy (uncertainty) and cannot separate the data into more homogenous subsets.
- This conclusion is achievable if we calculate entropy **(**$\mathbf{E_{feature1} < E_{feature2}}$**).**

Decision Trees, Example

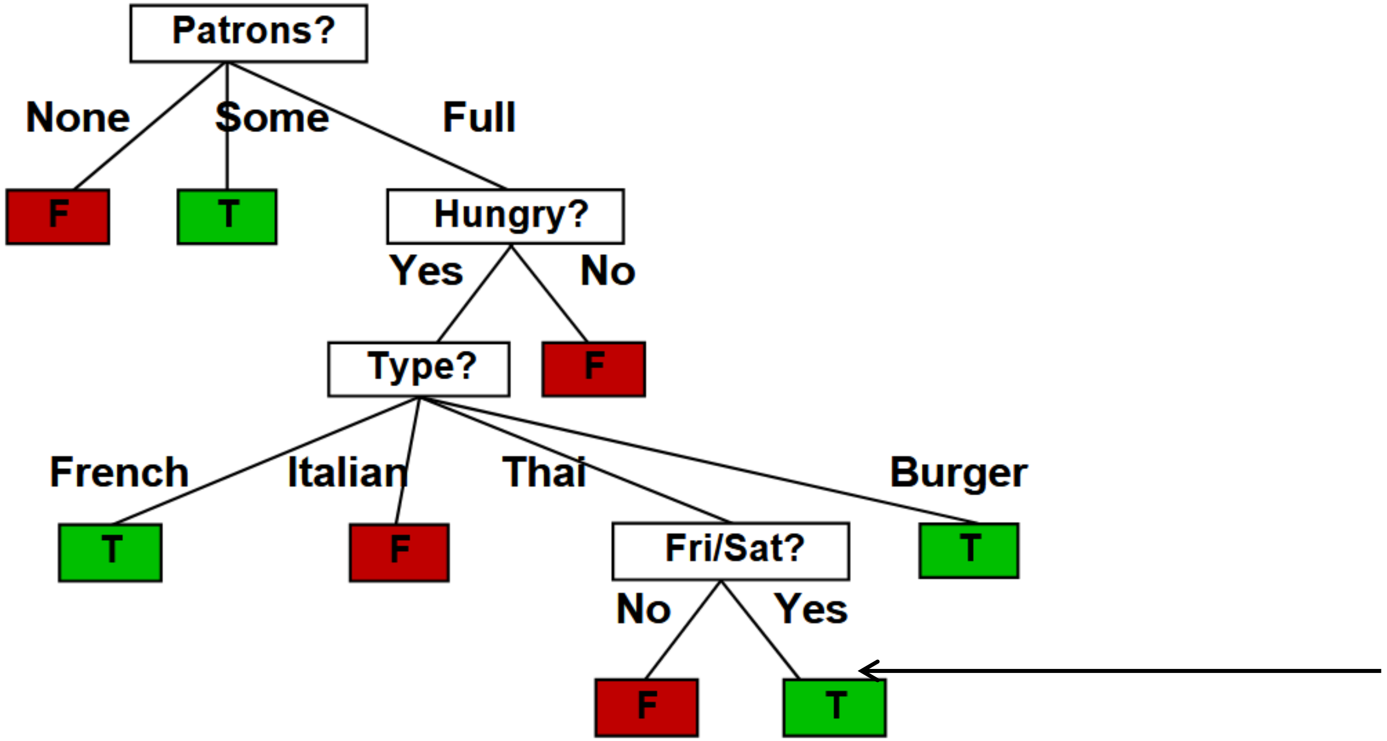


(a)

(b)

After splitting on Patrons, split the node Patrons=Full on Hungry

Decision Trees, Induced Decision Tree



- Does this leaf provide significant information?
- Is it worth keeping this leaf, or would it be better to prune it to reduce the complexity of the tree?

Decision Trees, Minimal Error Pruning

Following Ockham's Razor, ***prune*** branches that do not provide much benefit in classifying the items (aids generalization, avoids overfitting).

For a leaf node, all items will assign the majority class at that node. Estimate error rate on the (unseen) test items using the **Laplace error :**

$$E = 1 - \frac{n+1}{N+k}$$

N = total number of (training) items at the node

n = number of (training) items in the majority class

k = number of classes

If ***the average Laplace error of the children exceeds that of the parent node, prune off the children***.

Decision Trees, Minimal Error Pruning

Should the children of this node be pruned or not?

k=2, one is yes and the other is no

Left and Middle child have class frequencies [15,1]

$$E= 1-\frac{n+1}{N+k} = 1 - \frac{15+1}{16+2} = 0.111$$

Left and middle child have E= 0.111

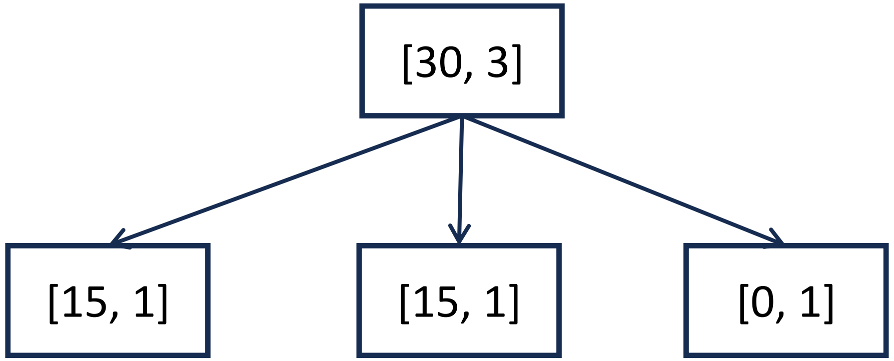
Right child has E = 0.333

Parent node has E = 4/35 = 0.114

Average for Left, Middle and Right child is:

$$E = \frac{16}{33}(0.111)+ \frac{16}{33}(0.111)+ \frac{1}{33}(0.333) = 0.118$$

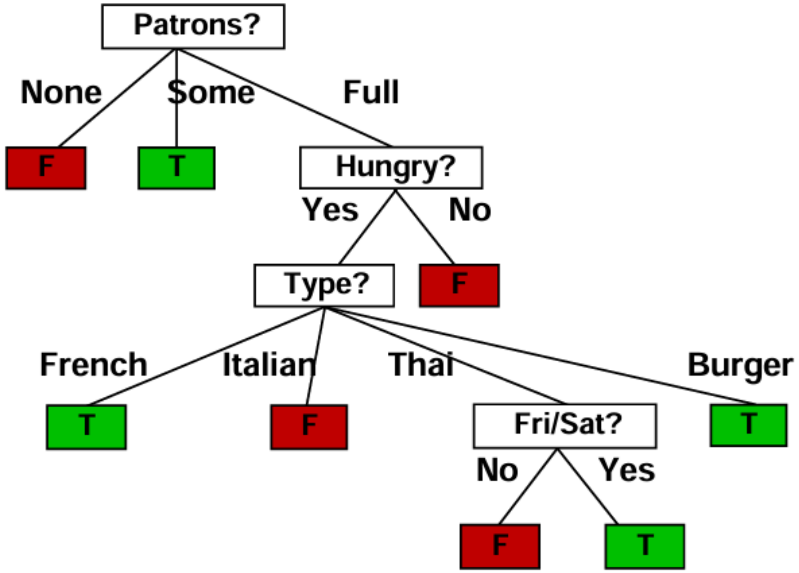
Since 0.118 > 0.114, children should be pruned



Decision Trees

Summary:

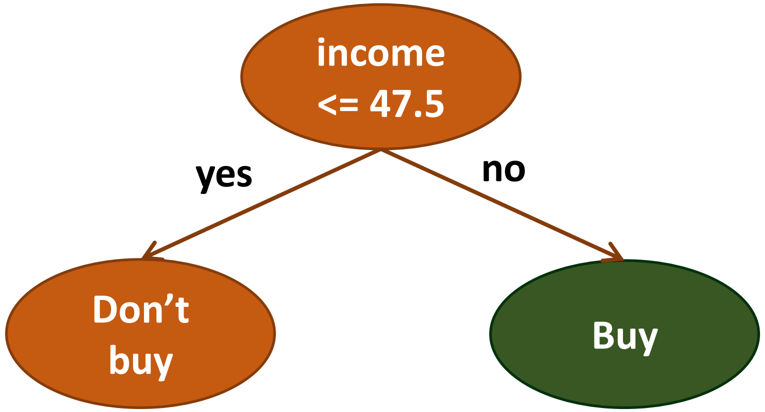
- A decision tree is a *supervised* learning algorithm used for both classification and regression tasks.
- It has a hierarchical, tree-like structure consisting of a **root node**, **branches**, **internal nodes**, and **leaf nodes**.
- The root node represents the entire dataset, and the tree splits data into subsets based on **specific features**, leading to decisions or predictions at the leaf nodes.



Decision Trees

Person	Income ($k)	Buys House (Yes=1 / No=-1)
A	35	-1
B	40	-1
C	45	1
D	50	1
E	55	1
F	60	1
G	65	-1
H	70	-1

Decision tree 1

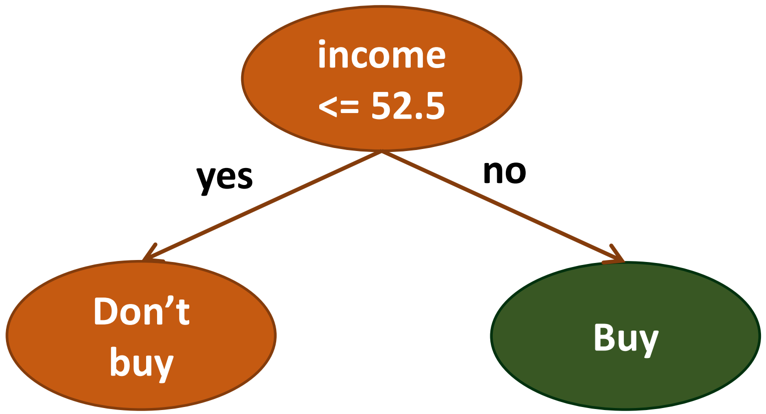


Decision Trees

Now imagine we **randomly remove Person G (65k, No)** from the training set. Without that data point, the split changes to the following.

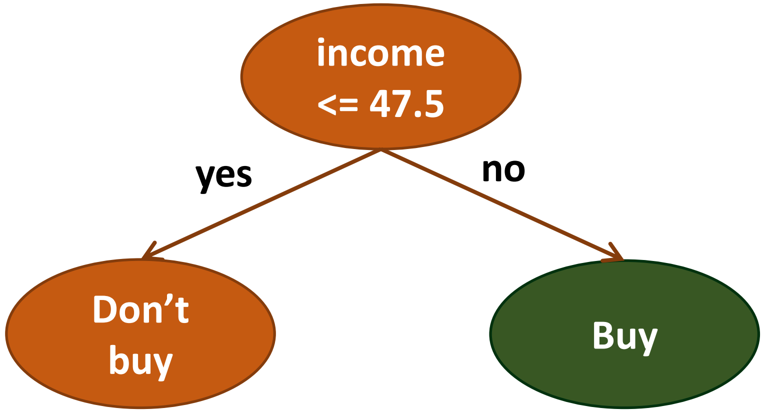
Person	Income ($k)	Buys House (Yes=1 / No=-1)
A	35	-1
B	40	-1
C	45	1
D	50	1
E	55	1
F	60	1
H	70	-1

Decision tree 2

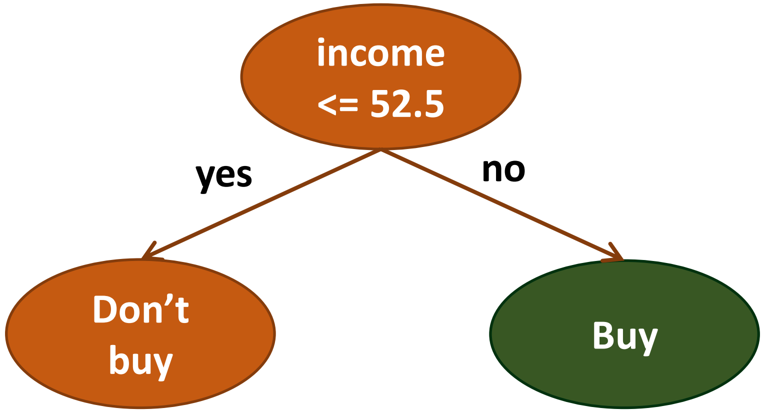


Decision Trees

Decision tree 1



Decision tree 2



- For example, consider a person with an income of $50k. The first decision tree predicts *Yes*, while the second tree predicts *No*.
- Such differences can arise from a tiny change in the training data (e.g., removing just one sample), leading to completely different predictions. This phenomenon is called *instability*, or ***high variance***.
    - But what is the solution?

Tree Limitations:

Models based on single trees have particular weaknesses.

- **Model instability:** Small changes in the data can drastically alter the structure of the tree or rules, leading to **high variance**.
- **Suboptimal predictive performance:** Single trees often fail to achieve strong **generalization**.

When training a decision tree, the dataset is typically divided into three subsets: training, validation, and test.

- Training set: Used to train the decision tree model.
- Validation set: Used to evaluate different configurations (hyperparameters) during training.
- Test set: Provides an unbiased evaluation of the final decision tree model on unseen data.

Generalization refers to the ability of a machine learning model to perform well on previously unseen data, not just the data it was trained on.

Tree Limitations:

Models based on single trees have particular weaknesses.

- **Model instability:** Small changes in the data can drastically alter the structure of the tree or rules, leading to **high variance**.
- **Suboptimal predictive performance:** Single trees often fail to achieve strong **generalization**.

Variance measures how much a model's predictions would change if it were trained on different datasets drawn from the same distribution.

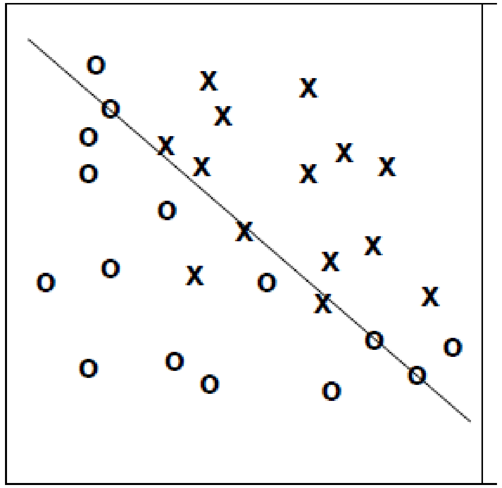
- **High variance** → The model tends to "memorize" the training data **(overfitting)** rather than learning the underlying patterns.
- **Low variance** → The model is more stable and consistent across different datasets.

When a model has high variance, it becomes overly sensitive to the specific training data. As a result, it may achieve excellent performance on the training set but generalize poorly to unseen data.

Tree Limitations:

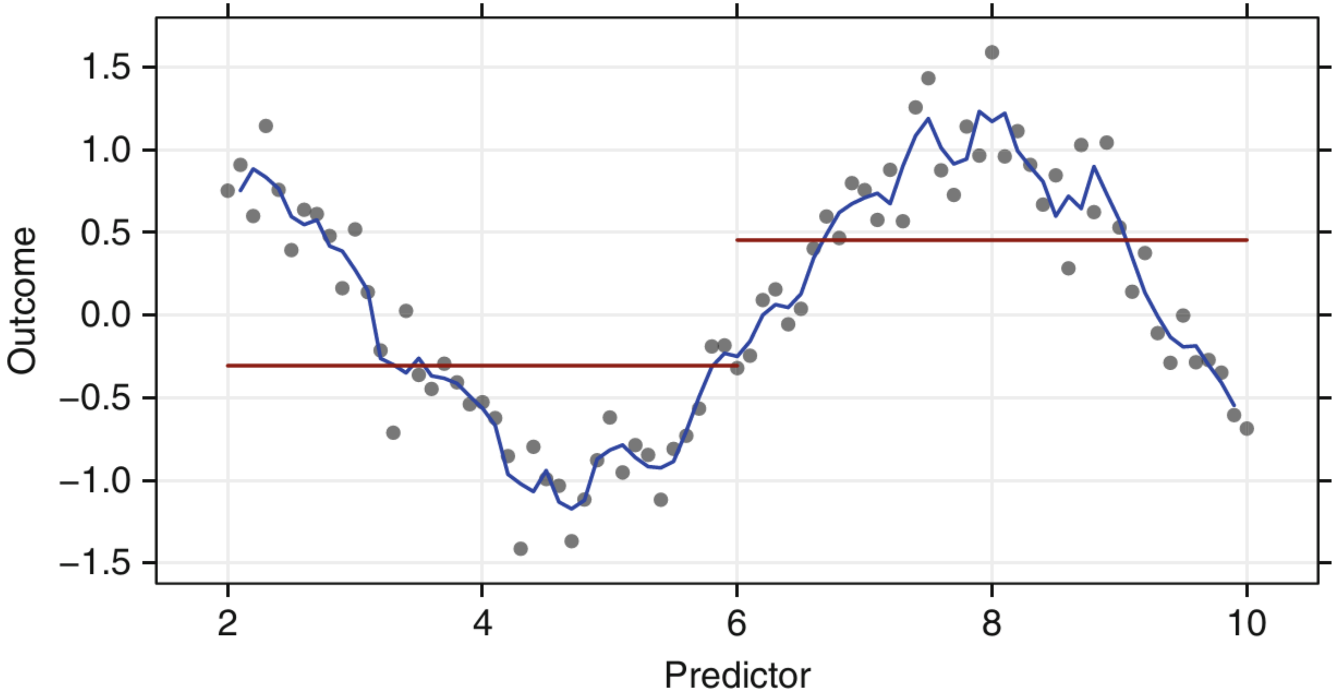
- Bias refers to the **systematic error** introduced by approximating a real-world problem (which may be very complex) with a simpler model.

- A **high bias model** makes **strong assumptions** about the data, leading to **underfitting**.
- It can’t capture the true complexity of the underlying relationship.



Tree Limitations:

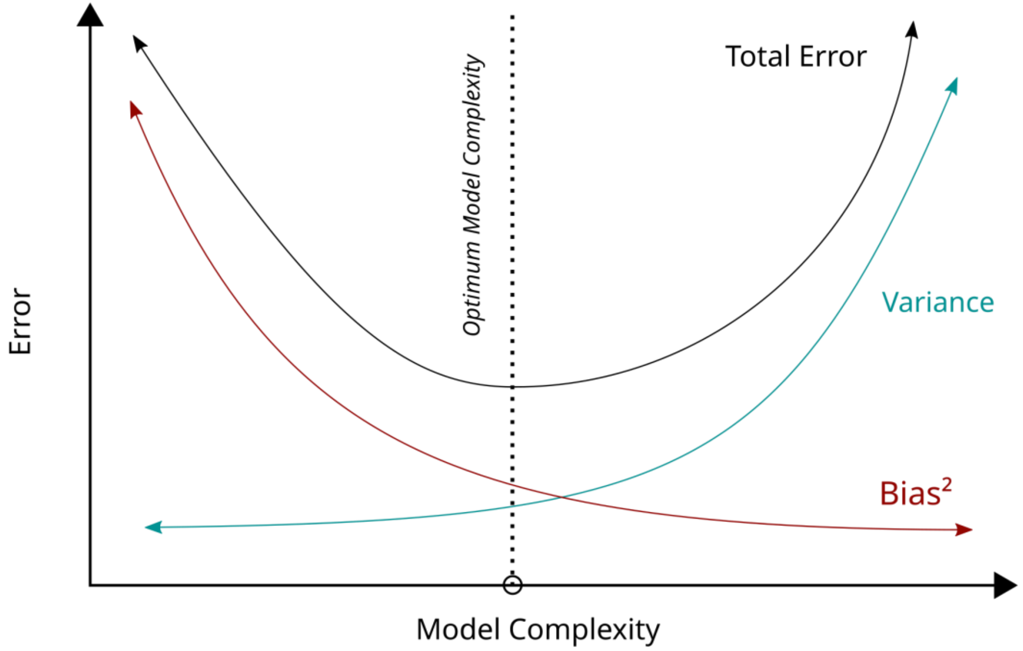
- Two model fits to a sin wave. It shows extreme examples of models that are either high bias or high variance. The red line predicts the data using simple averages of the first and second half of the data. The blue line is a three-point moving average.



Tree Limitations:

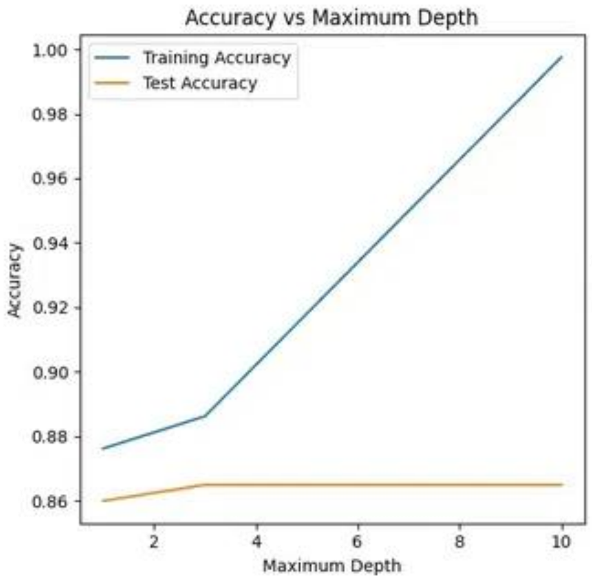
- A complex model would help us to avoid bias (underfitting) and limiting the model complexity would help us to avoid overfitting (high variance).

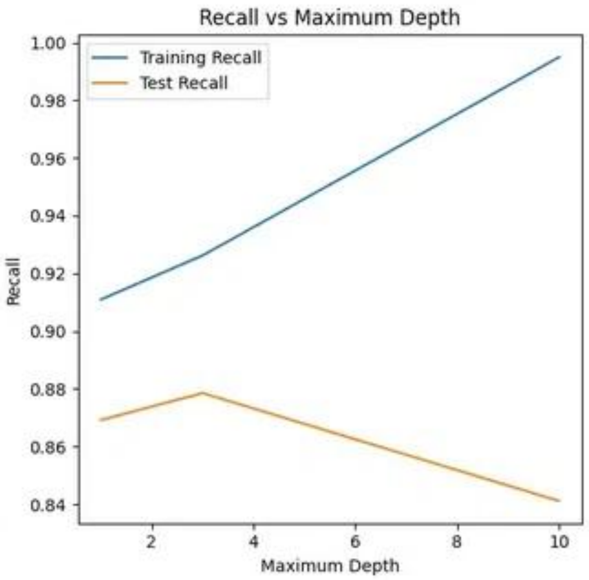
- There is a trade-off here as it is known as bias-variance tradeoff.



Tree Depth:

- The depth of a tree is the length of the **longest path** from the root node to any leaf node.
  - Shallow trees suffer from high bias (underfitting) because the tree does not have enough nodes to capture the complexity of the data.
  - Deep trees suffer from high variance (overfitting) because the tree may assign a unique path to each data sample rather than learning general patterns.
- The optimal depth should be determined based on factors such as dataset size, data complexity, and available hardware resources.





Ensemble Learning with Trees:

Models based on single trees or rules have particular weaknesses.

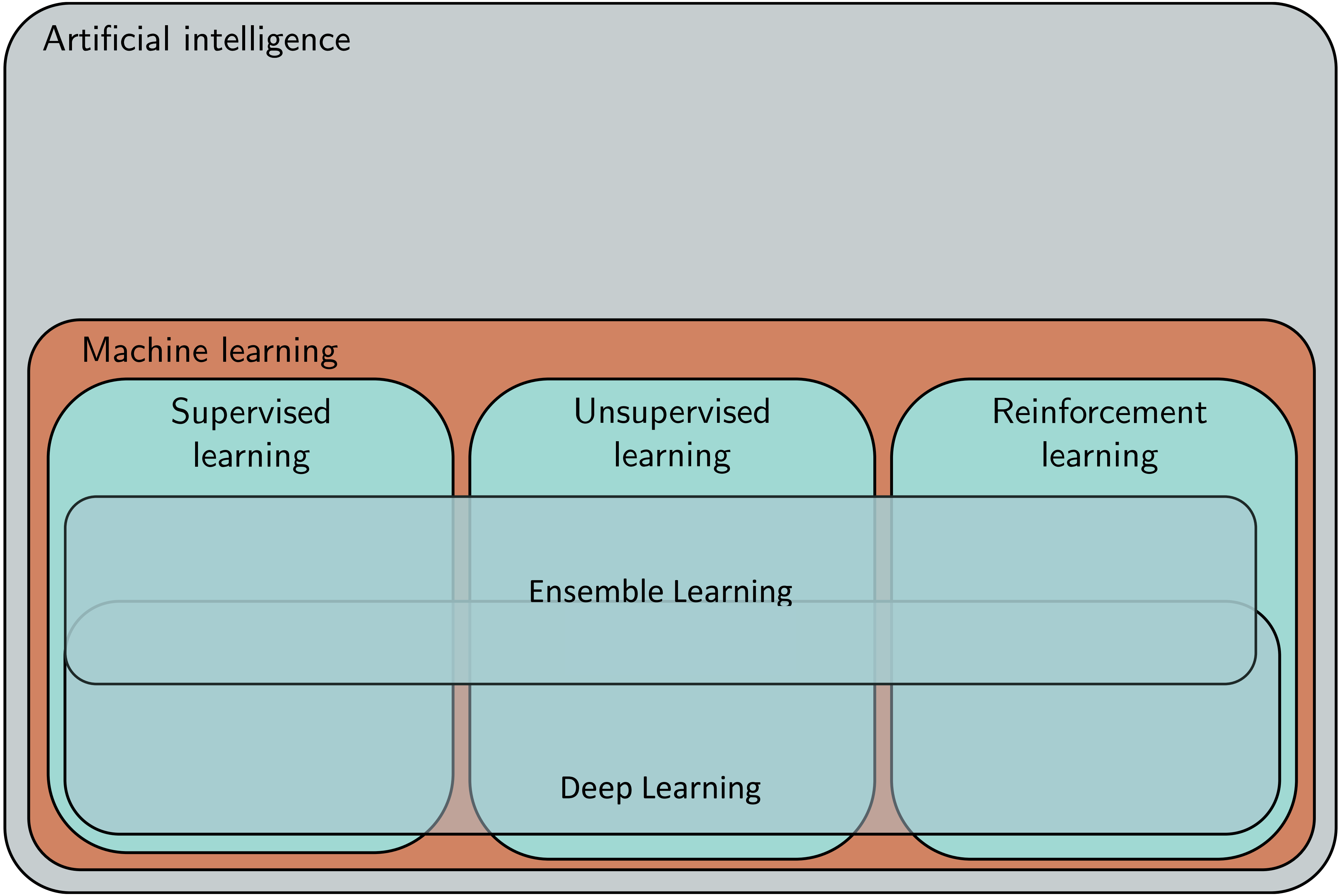
- ➢ **Model instability:** slight changes in the data can drastically alter the structure of the tree or rules (high variance).
- ➢ **Less-than-optimal predictive performance.**

- To address these issues, researchers developed **ensemble methods**, which combine many trees (or other machine learning models) into a single, more robust model.

- While ensemble learning can refer to the combination of various machine learning approaches, in this lecture we will focus specifically on **tree-based ensemble methods**.

- Ensembles generally provide much better predictive performance than individual models.

- ✓ Different Subfields of AI Algorithms.
- ✓ Decision Trees.
- ✓ **Ensemble Learning:**
  - Bagged Trees.
  - Random Forests.
  - Boosting Trees.


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Ensemble Learning
Deep Learning

Ensemble Learning with Trees:

Models based on single trees or rules, do have particular weaknesses.

- **Model instability**
- **Less-than-optimal predictive performance**.

To address these problems, **ensemble techniques**—methods that combine the predictions of many models—emerged in the 1990s.

Ensembles generally achieve much better predictive performance than single trees, and this is often true for other types of models as well.

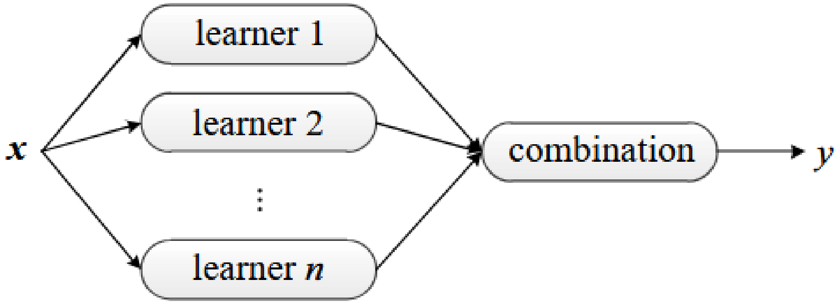


Fig. 2: A common ensemble architecture.