Machine & Deep Learning

Pr Rahhal ERRATTAHI
rahhal.errattahi@um6p.ma
errattahi.r@ucd.ac.ma

Lecture 01

Course Topics

- Part 1: Introduction to ML
- Part 2: Supervised ML problem setup and Data Preprocessing
- Part 3: Kernel Methods
- Part 4: Decision Trees
- Part 5: Regression
- Part 6: Ensemble learning
- Part 7: Supervised learning of Neural Networks

Lab sessions

- Python notebooks in CoLab
- Alternatively, set up Jupyter Notebooks yourself
  - Numpy
  - Matplotlib
  - scikit learn
  - tensorflow
  - PyTorch

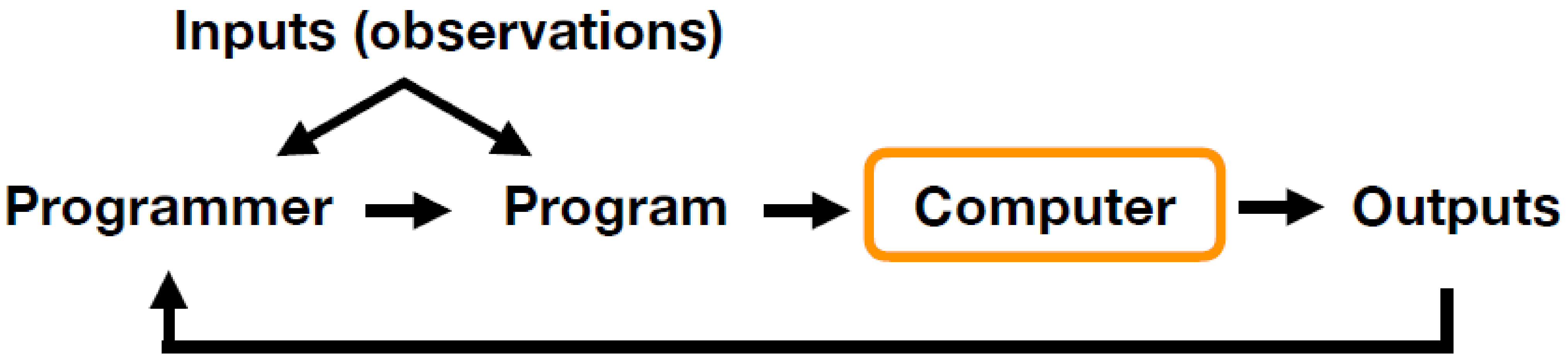
Lecture 1 Overview

- What is machine learning ?
- Categories of machine learning
- Notation
- Approaching a machine learning application
- ML Terminology

Lecture 1 Overview

- What is machine learning ?
- Categories of machine learning
- Notation
- Approaching a machine learning application
- ML Terminology

The Traditional Programming Paradigm



What is Machine Learning?

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed"

-- Arthur L. Samuel, AI pioneer, 1959

What is Machine Learning?

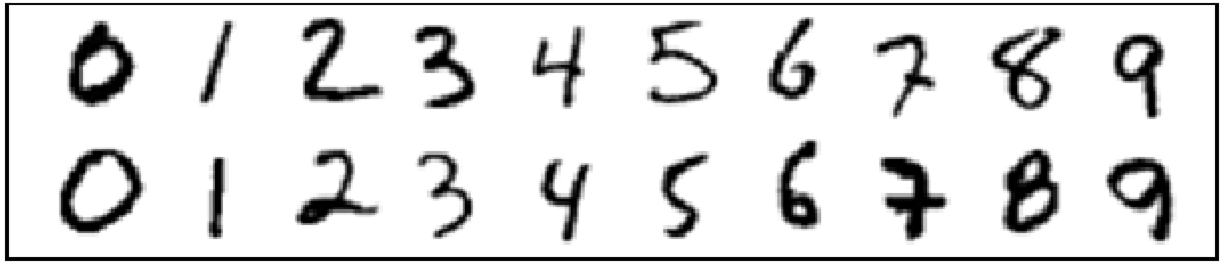
- Machine learning algorithms are algorithms that **learn** models from data / experience.
- No need to formulate explicit rules.
- Algorithm performance gets better with experience / data.

Defining the Learning Task

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

— Tom Mitchell, Professor at Carnegie Mellon University

Handwriting Recognition Example:



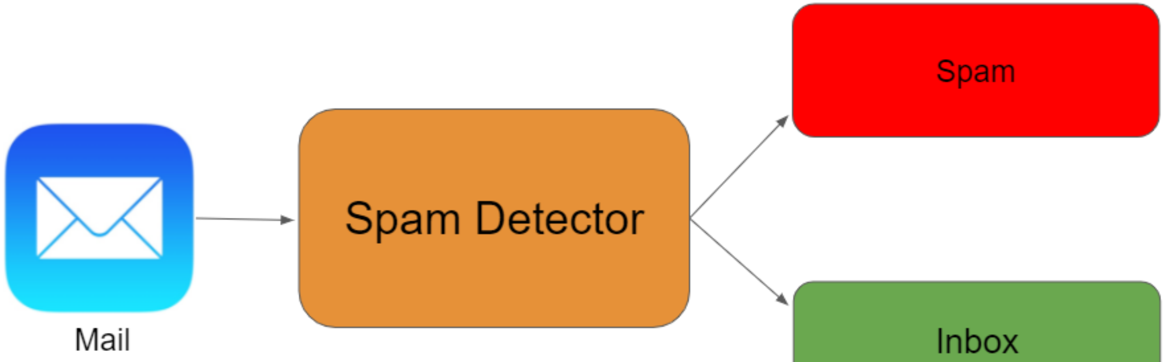
Task T: ?

Performance measure P: ?

Training experience E: ?

Defining the Learning Task

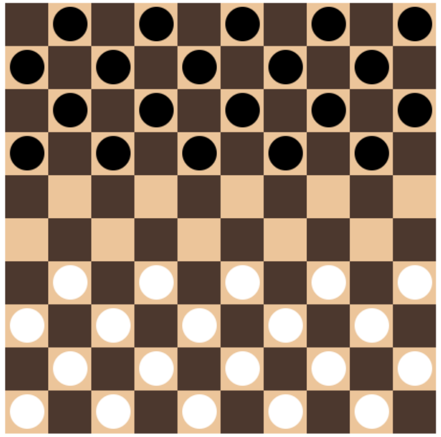
Improve on task T, with respect to performance metric P, based on experience E



T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

Defining the Learning Task

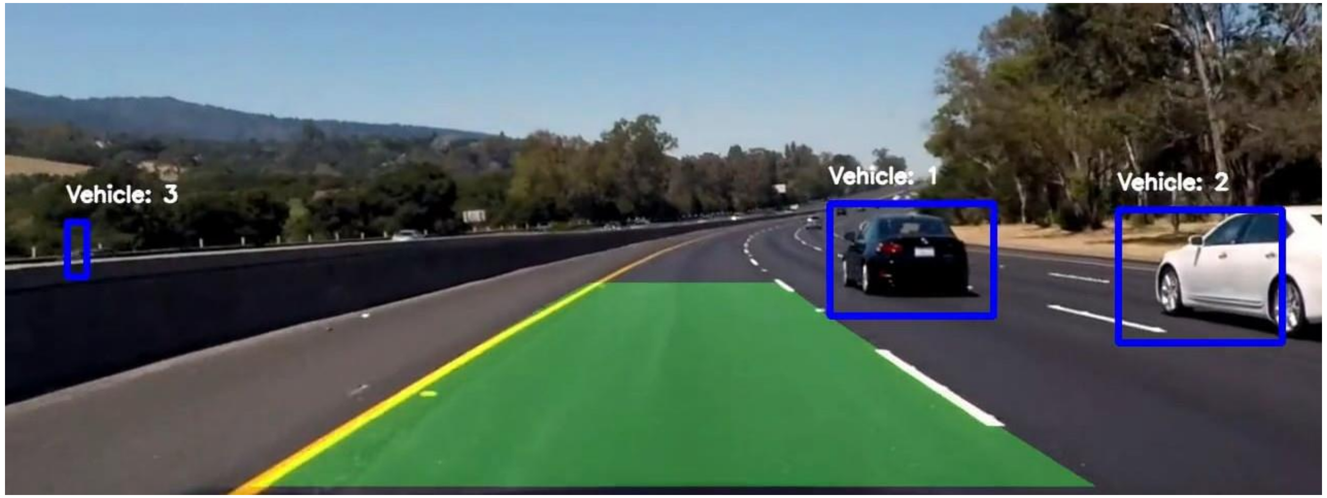
Improve on task T, with respect to performance metric P, based on experience E



T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

Defining the Learning Task

Improve on task T, with respect to performance metric P, based on experience E



T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can’t explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)

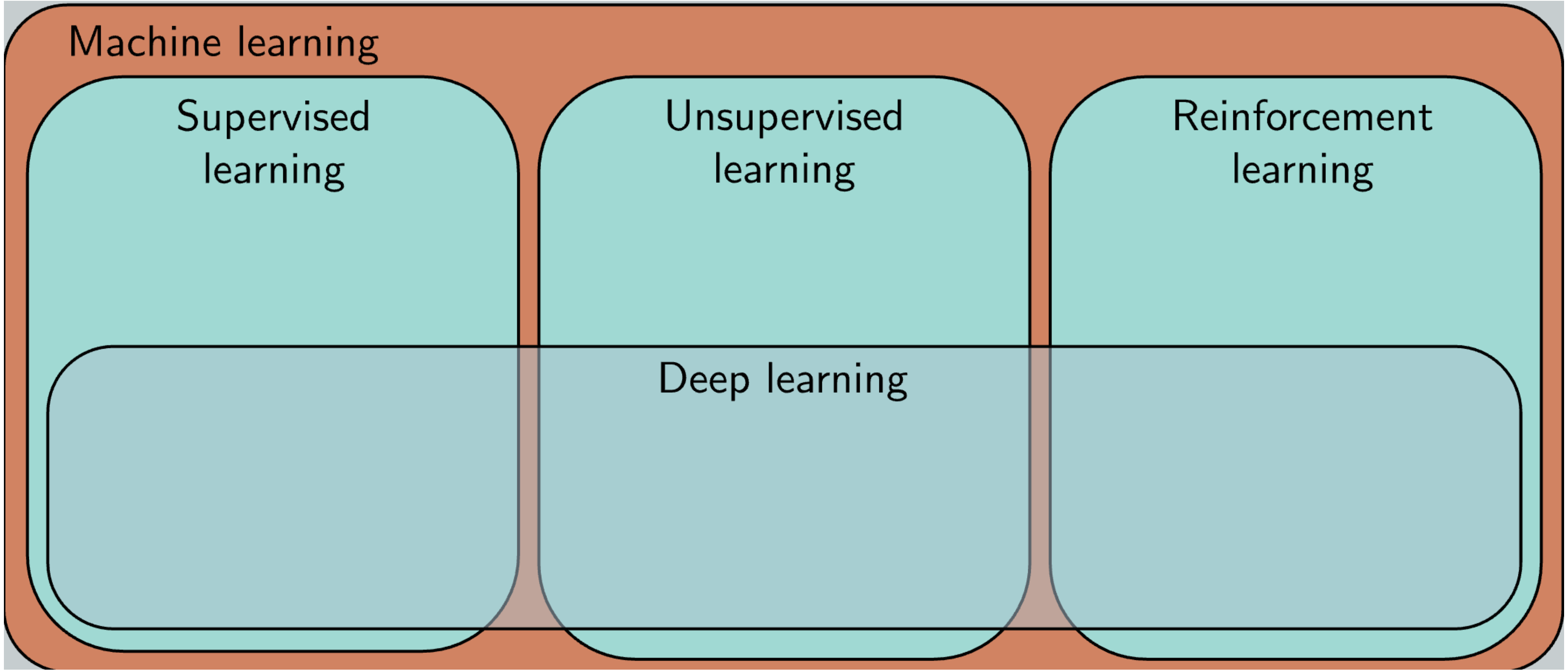
Learning isn’t always useful:

- There is no need to “learn” to calculate payroll

Lecture 1 Overview

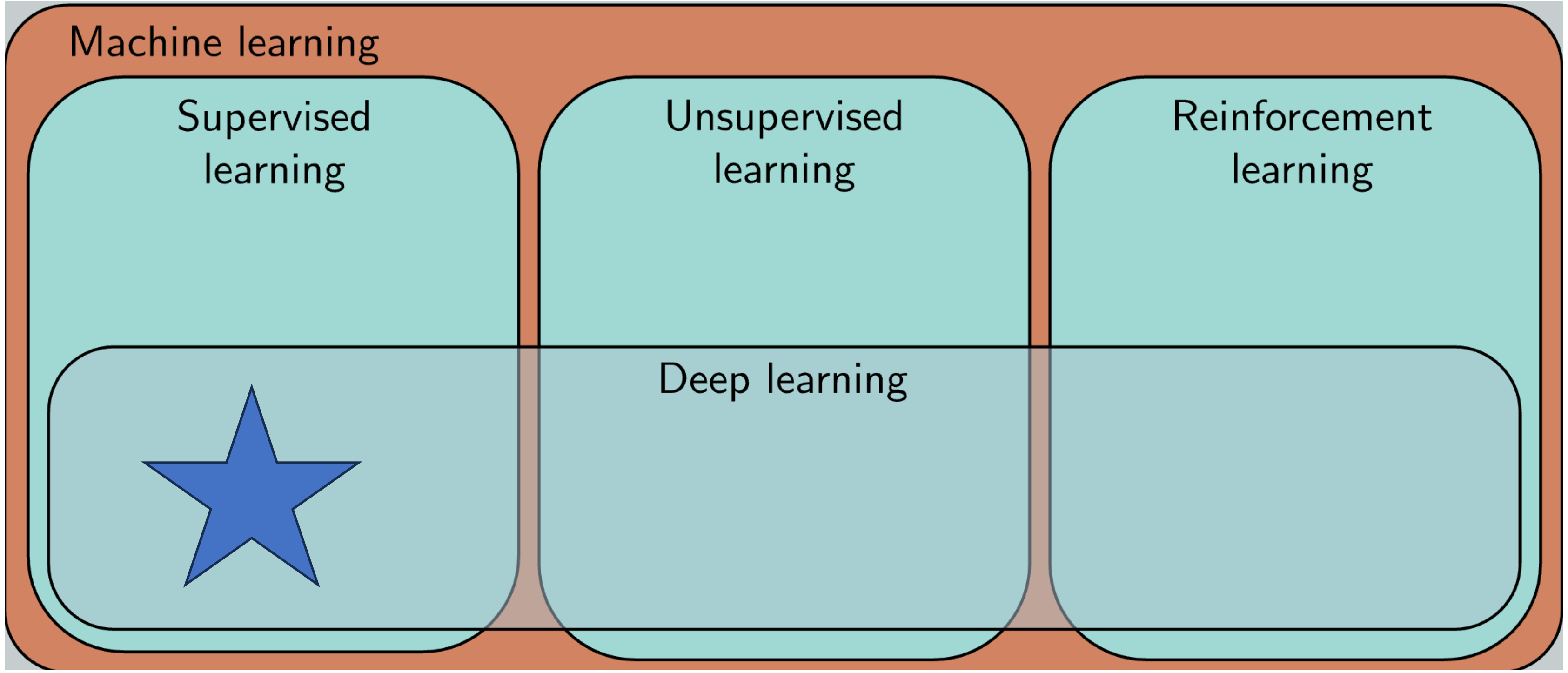
- What is machine learning ?
- Categories of machine learning
- Notation
- Approaching a machine learning application
- ML Terminology

Categories of machine learning

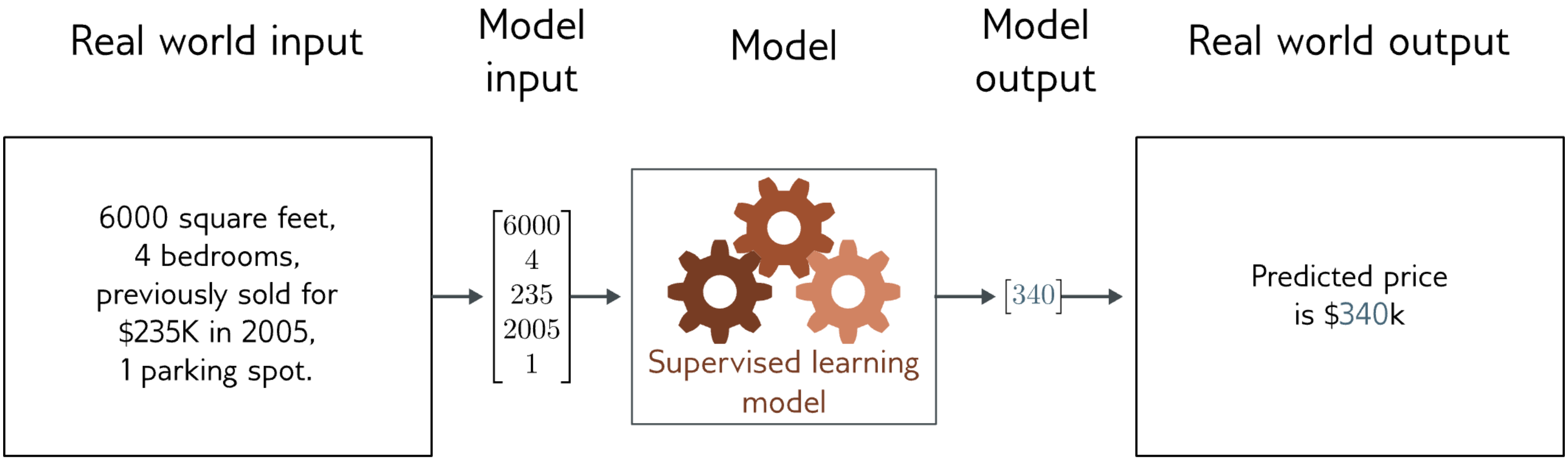


Supervised learning

- Define a mapping from input to output
- Learn this mapping from paired input/output data examples

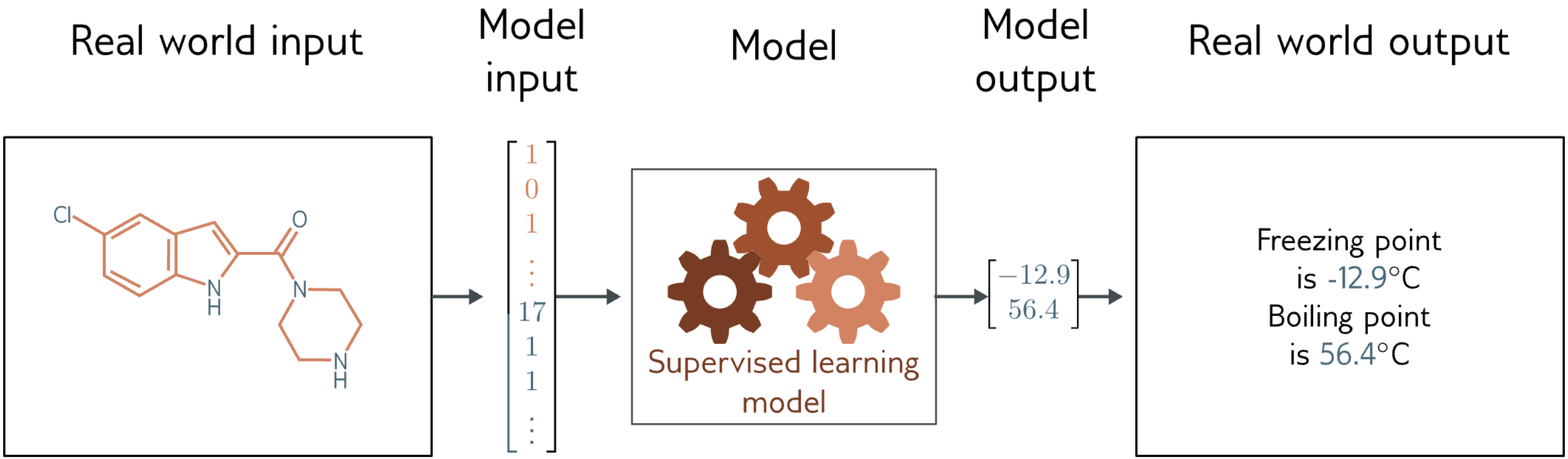


Regression



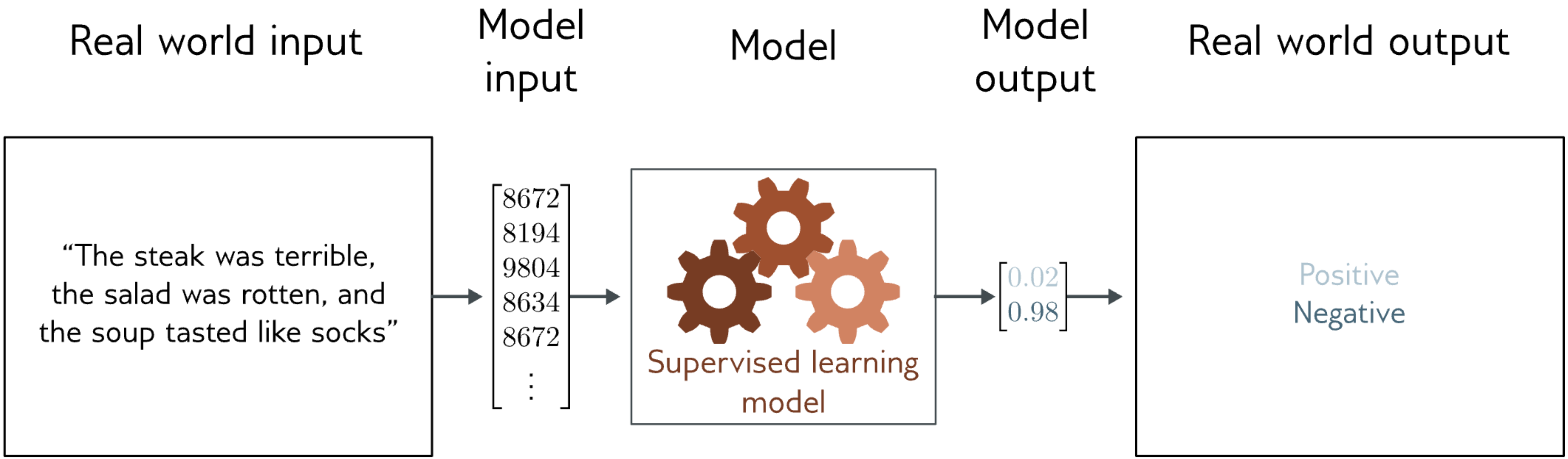
- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression



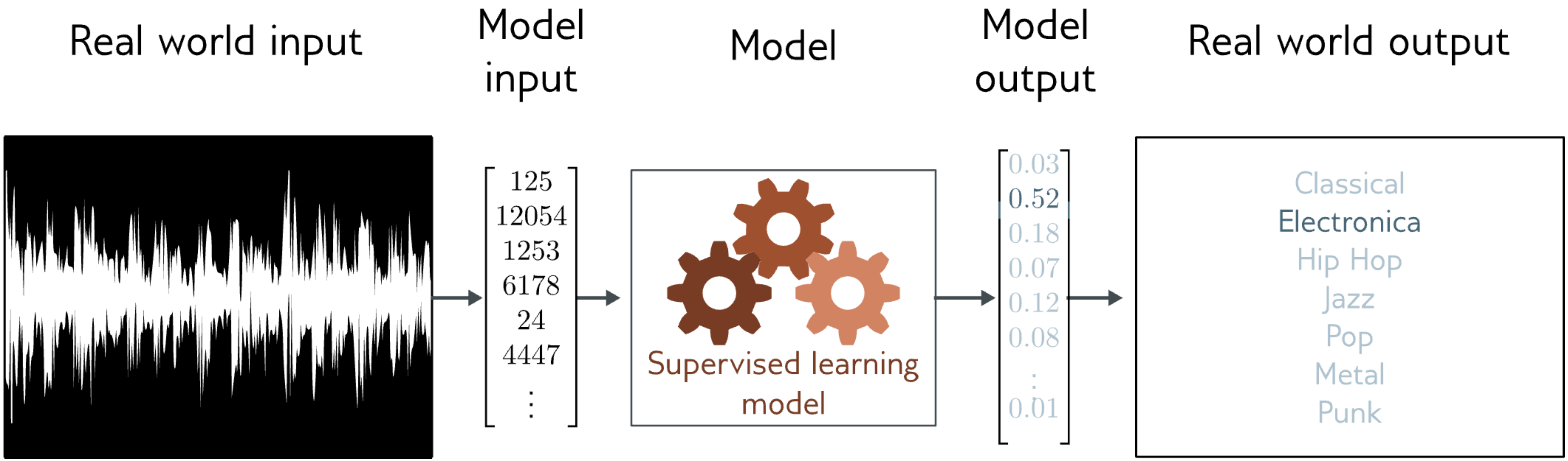
- Multivariate regression problem (>1 output, real value)
- Graph neural network

Text classification



- Binary classification problem (two discrete classes)
- Transformer network

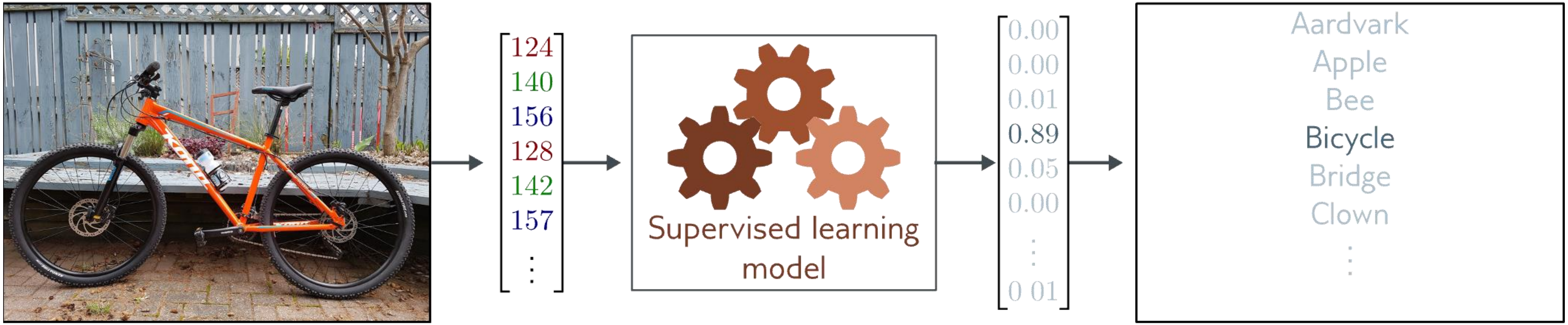
Music genre classification



- Multiclass classification problem (discrete classes, >2 possible values)
- Recurrent neural network (RNN)

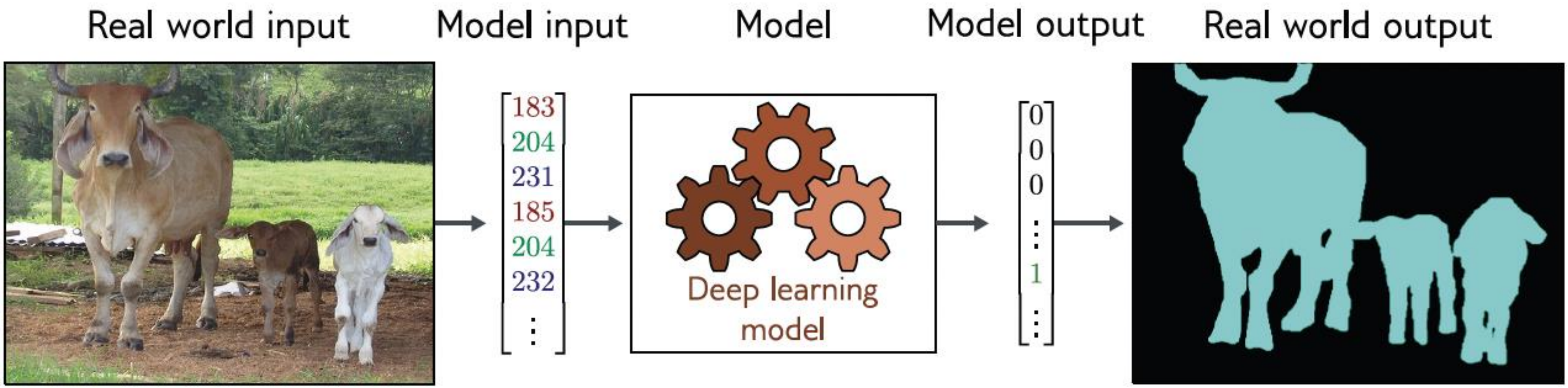
Image classification

Real world input | Model input | Model | Model output | Real world output



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

Image segmentation



- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

ML Terminology

- **Regression** = continuous numbers as output
- **Classification** = discrete classes as output
- **Two class** and **multiclass** classification treated differently
- **Univariate** = one output
- **Multivariate** = more than one output

Translation

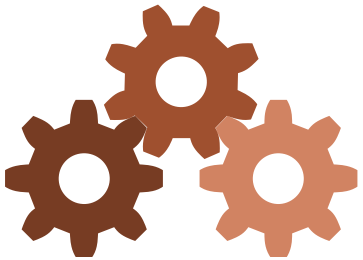
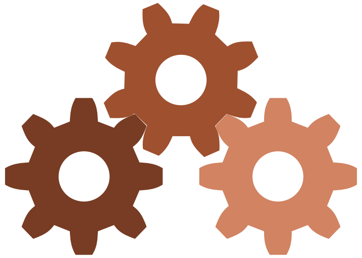
Real world input	Model input	Model	Model output	Real world output
"Skill without imagination is craftsmanship and gives us many useful objects such as wickerwork picnic baskets. Imagination without skill gives us modern art."	$\begin{bmatrix}7800\\9853\\4520\\4596\\987\\\vdots\end{bmatrix}$	 Supervised learning model	$\begin{bmatrix}6003\\3689\\4432\\6003\\2149\\\vdots\end{bmatrix}$	"L'habileté sans l'imagination est de l'artisanat et nous donne de nombreux objets utiles tels que des paniers de pique-nique en osier. L'imagination sans habileté nous donne l'art moderne."

Image captioning

Real world input | Model input | Model | Model output | Real world output

$$\begin{bmatrix} 183 \\ 204 \\ 231 \\ 185 \\ 204 \\ 232 \\ \vdots \end{bmatrix}$$

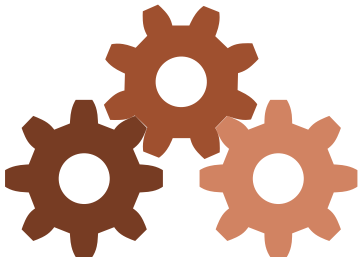


Supervised learning model

$$\begin{bmatrix} 1 \\ 5593 \\ 7532 \\ 7924 \\ 1 \\ \vdots \end{bmatrix}$$

"A Kazakh man on a horse holding a bird of prey"

Image generation from text

Real world input	Model input	Model	Model output	Real world output
"Teddy bears mixing sparkling chemicals as mad scientists, in a steampunk style."	$\begin{bmatrix} 8300 \\ 532 \\ 7676 \\ 7898 \\ 883 \\ \vdots \end{bmatrix}$	 Supervised learning model	$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ \vdots \end{bmatrix}$	

What do these examples have in common?

- Very complex relationship between input and output
- Sometimes may be many possible valid answers
- But outputs (and sometimes inputs) obey rules

"A Kazakh man on a horse holding a bird of prey"

Language obeys grammatical rules

Natural images also have "rules"

Supervised learning

Supervised learning is about function approximation

Problem Setting:

- Set of possible instances X
- Unknown target function $f : X \rightarrow Y$
- Set of function hypotheses $H = \{h \mid h : X \rightarrow Y\}$

Input:

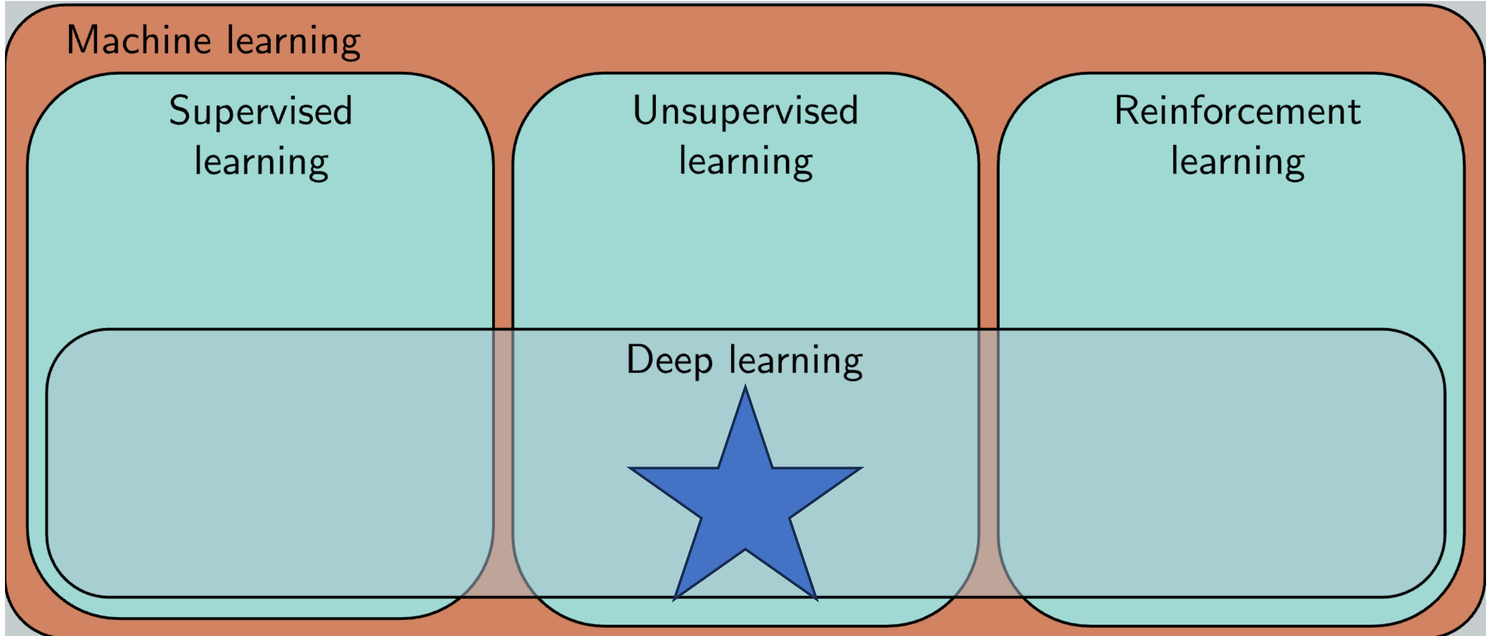
- training examples $\{< x_i, y_i >\}$. For example x is an email and y is either Spam or No Spam.

Output:

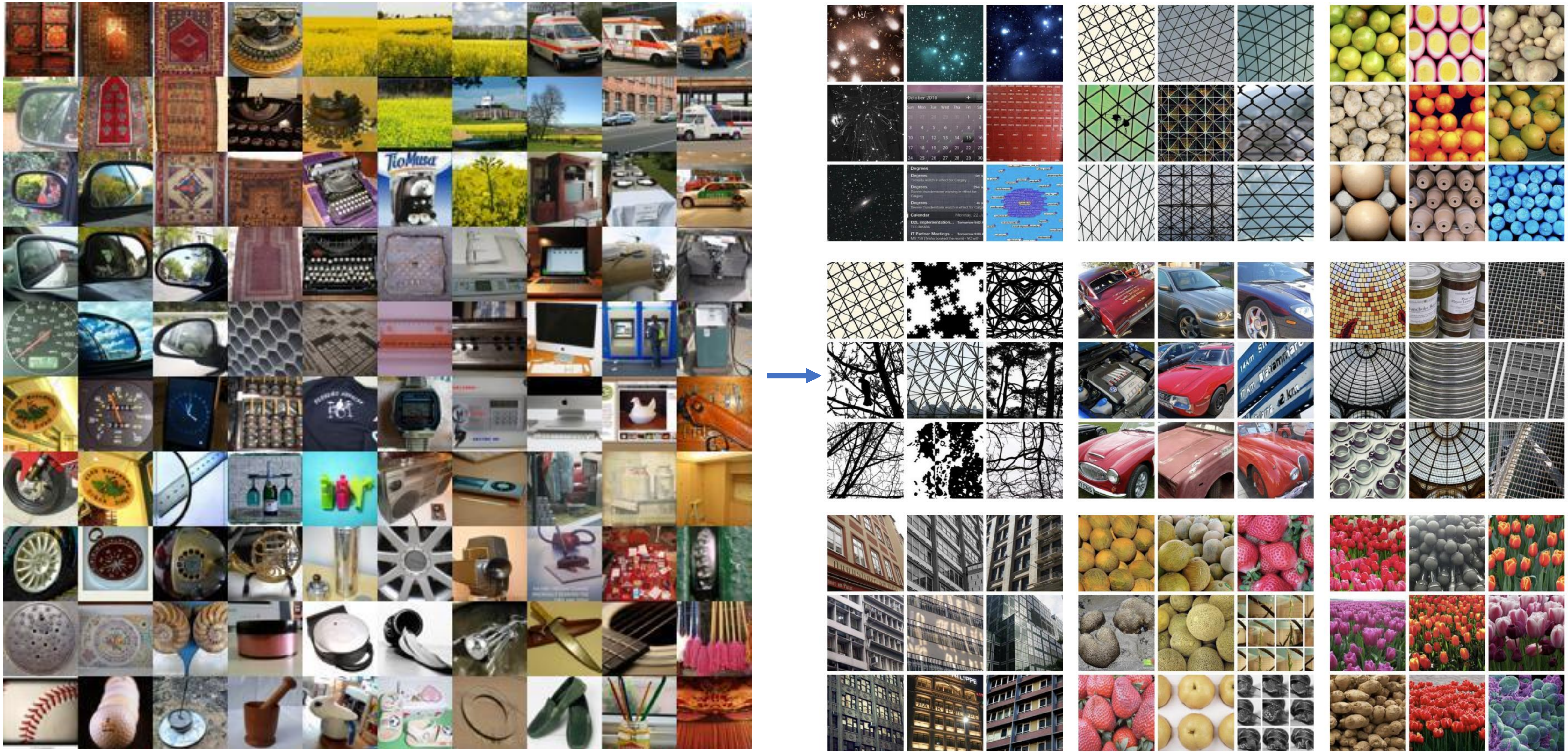
- Hypothesis $h \in H$ that best approximates target function f . OR
- a classification “rule” that can determine the class of any object from its attributes values.

Unsupervised Learning

- Unsupervised learning is about description, opposed to approximation (supervised learning).
  - Clustering
  - Finding outliers
  - Generating new examples
  - Filling in missing data



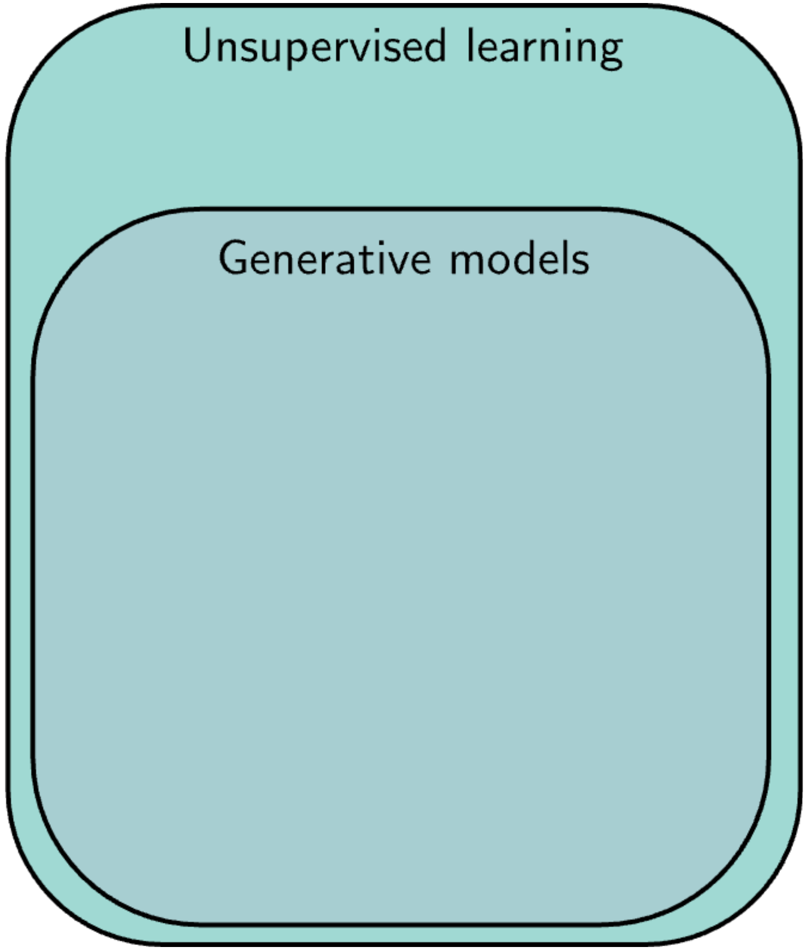
DeepCluster: Deep Clustering for Unsupervised Learning of Visual Features (Caron et al., 2018)



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

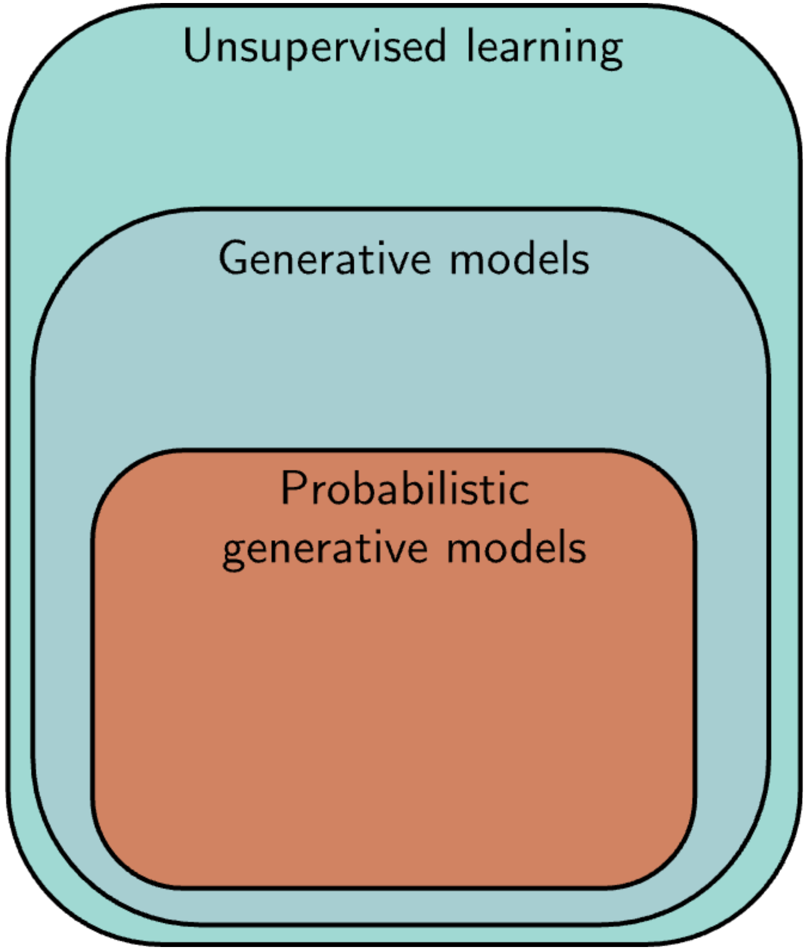
Unsupervised Learning

- Learning about a dataset without labels
  - e.g., clustering
- Generative models can create examples
  - e.g., generative adversarial networks

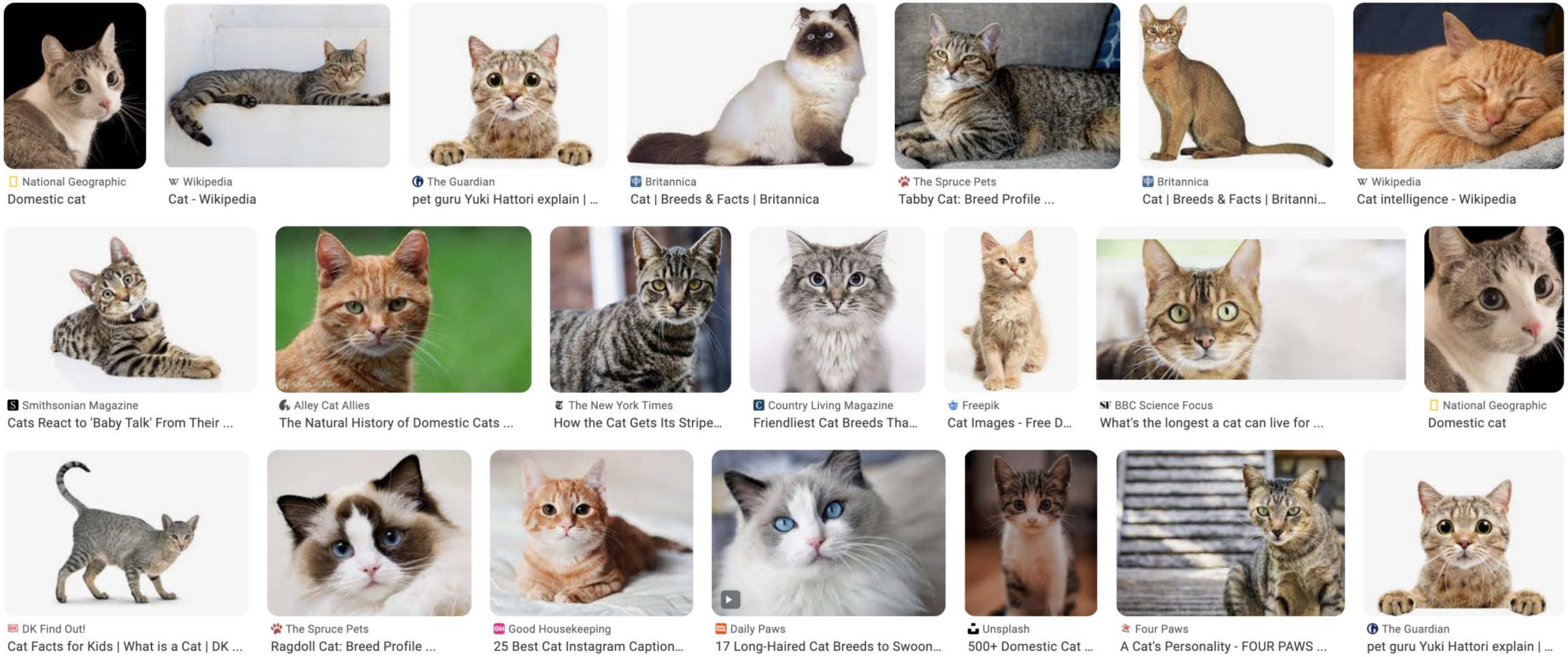


Unsupervised Learning

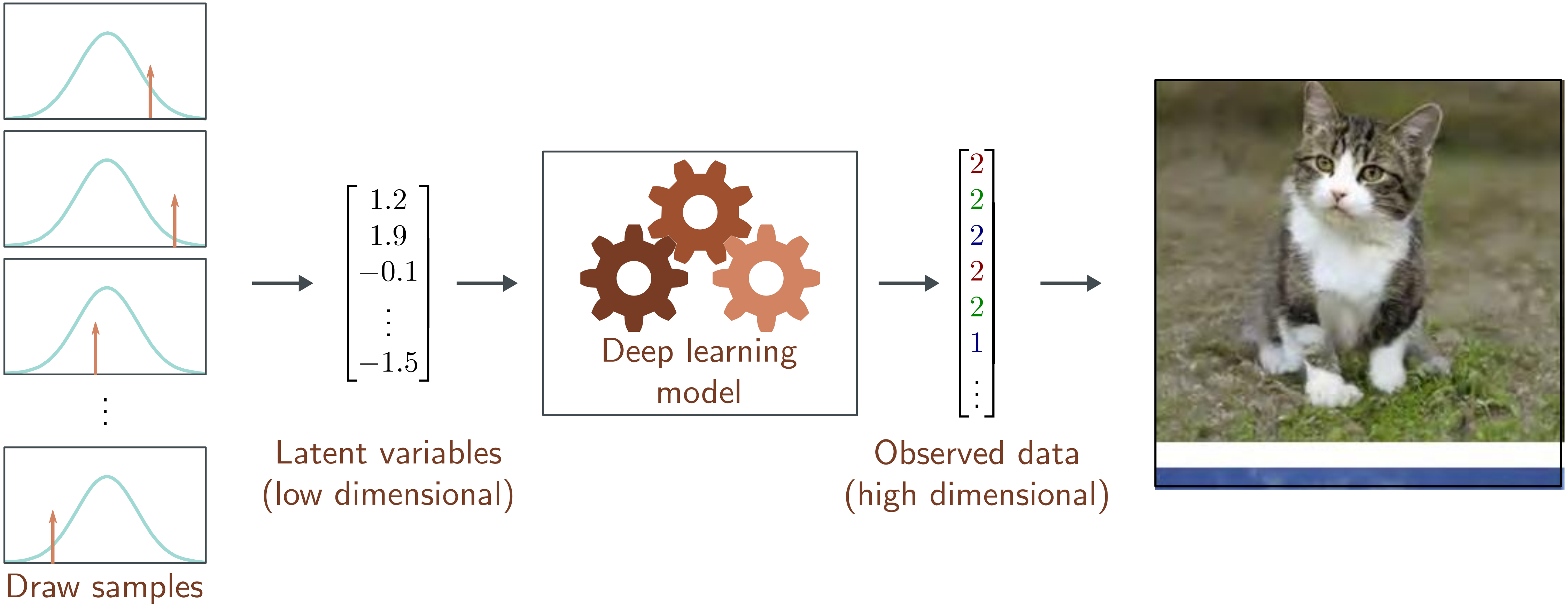
- Learning about a dataset without labels
    - e.g., clustering
- Generative models can create examples
    - e.g., generative adversarial networks
- PGMs learn distribution over data
    - e.g., variational autoencoders,
    - e.g., normalizing flows,
    - e.g., diffusion models



Generative models

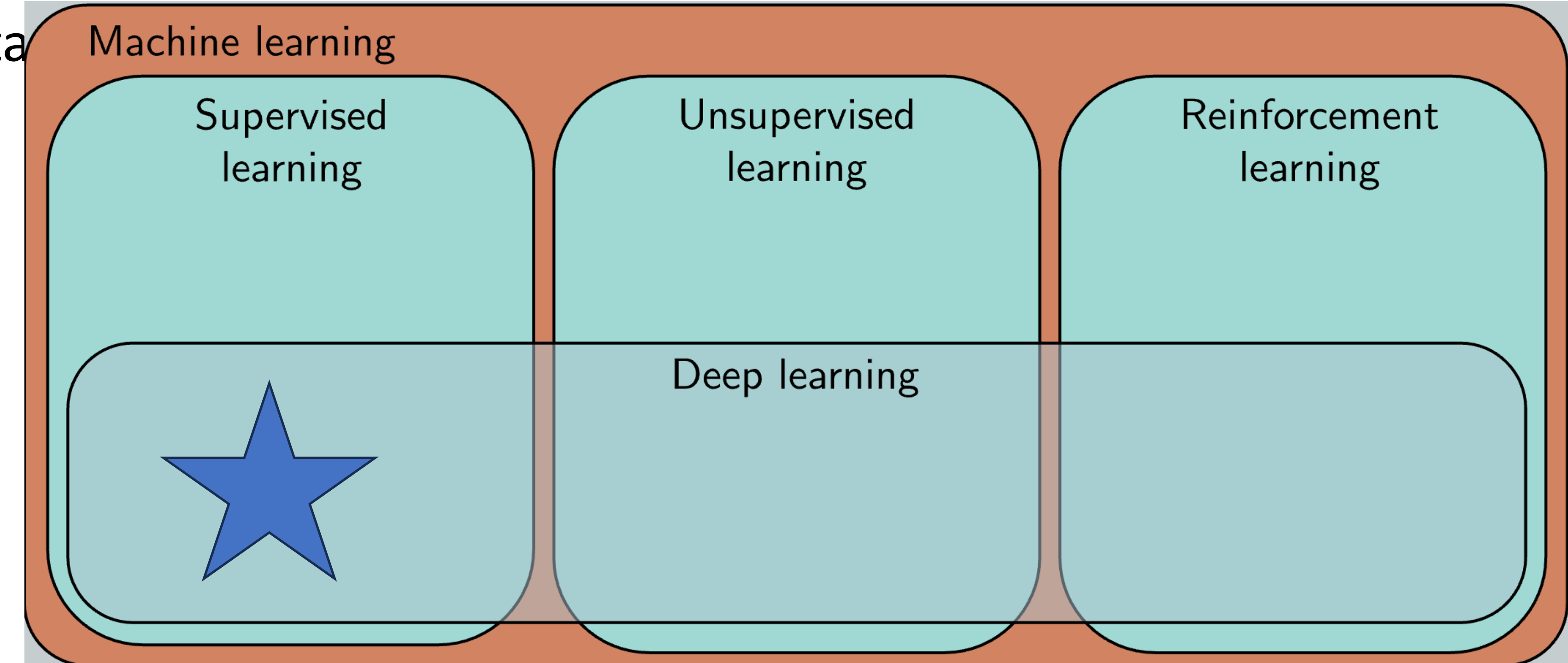


Latent variables



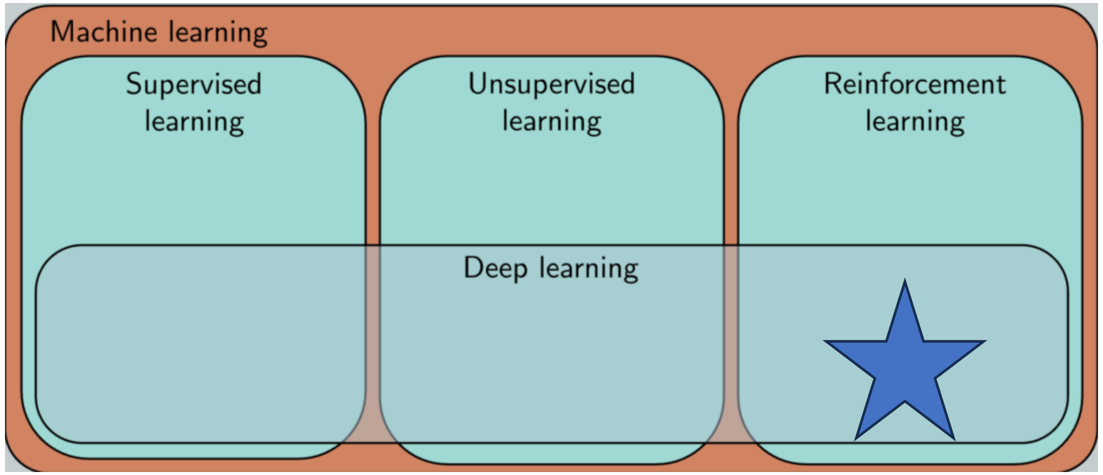
Unsupervised Learning

- Learning about a dataset without labels
  - Clustering
  - Finding outliers
  - Generating new examples
  - Filling in missing data



Reinforcement learning

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



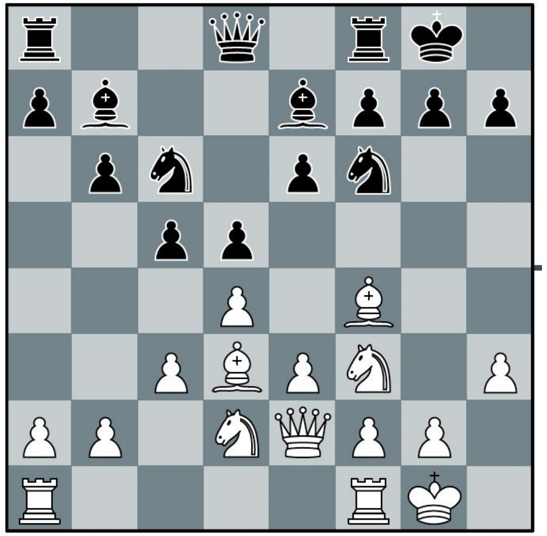
- Goal: take actions to change the state so that you receive rewards

- You don't receive any data – you have to explore the environment yourself to gather data as you go

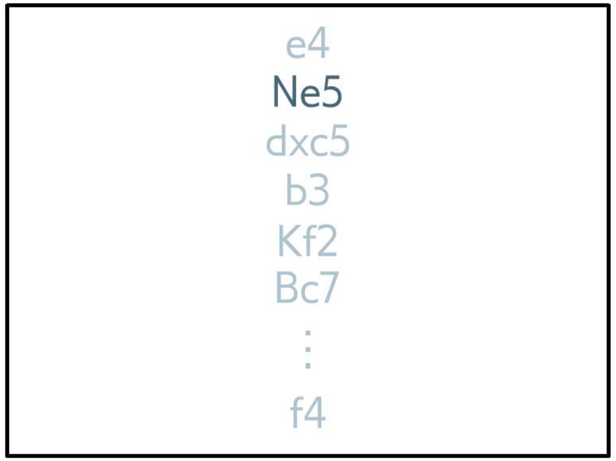
Example: chess

- States are valid states of the chess board
- Actions at a given time are valid possible moves
- Positive rewards for taking pieces, negative rewards for losing them

State

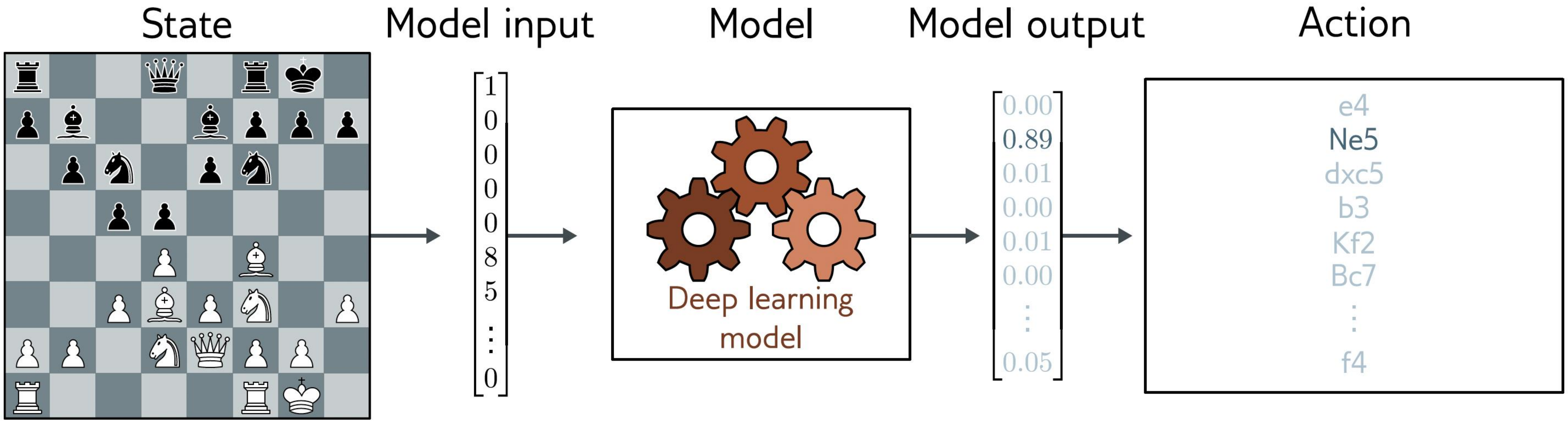


Action



Example: chess

- States are valid states of the chess board
- Actions at a given time are valid possible moves
- Positive rewards for taking pieces, negative rewards for losing them



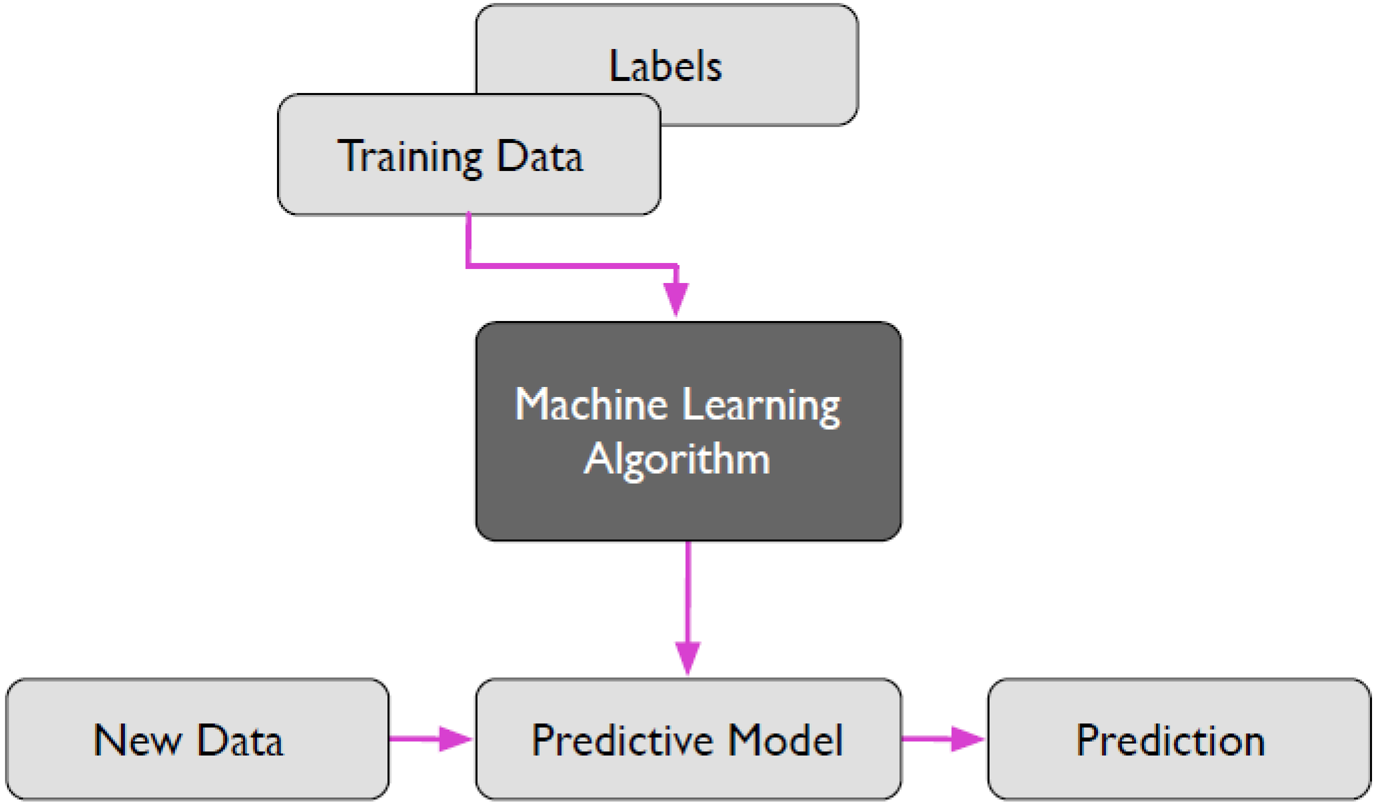
Why is this difficult?

- Stochastic
  - Make the same move twice, the opponent might not do the same thing
  - Rewards also stochastic (opponent does or doesn’t take your piece)
- Temporal credit assignment problem
  - Did we get the reward because of this move? Or because we made good tactical decisions somewhere in the past?
- Exploration-exploitation trade-off
  - If we found a good opening, should we use this?
  - Or should we try other things, hoping for something better?

Lecture 1 Overview

- What is machine learning ?
- Categories of machine learning
- Notation
- Approaching a machine learning application
- ML Terminology

Supervised Learning Workflow -- Overview



Supervised Learning Notation

- Training set: $\mathcal{D} = \{\langle x^{[i]}, y^{[i]} \rangle, i = 1, \dots, n\}$,
- Unknown function: $f(\boldsymbol{x}) = y$
- Hypothesis: $h(\boldsymbol{x}) = \hat{y}$

Classification

$$h : \mathbb{R}^m \rightarrow \_\_$$

Regression

$$h : \mathbb{R}^m \rightarrow \_\_$$

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Design matrix

$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix}$$

Design matrix

Data Representation

$m =$ __

$n =$ __

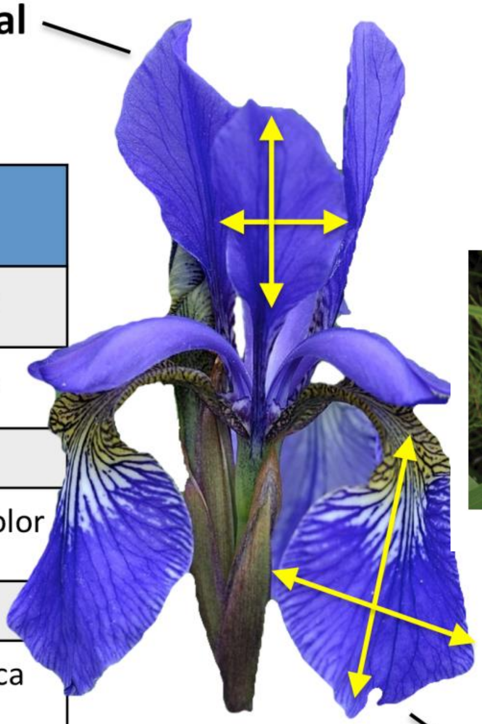
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)

Petal



Sepal

Iris flower data set

Iris Versicolor **Iris Setosa** **Iris Virginica**

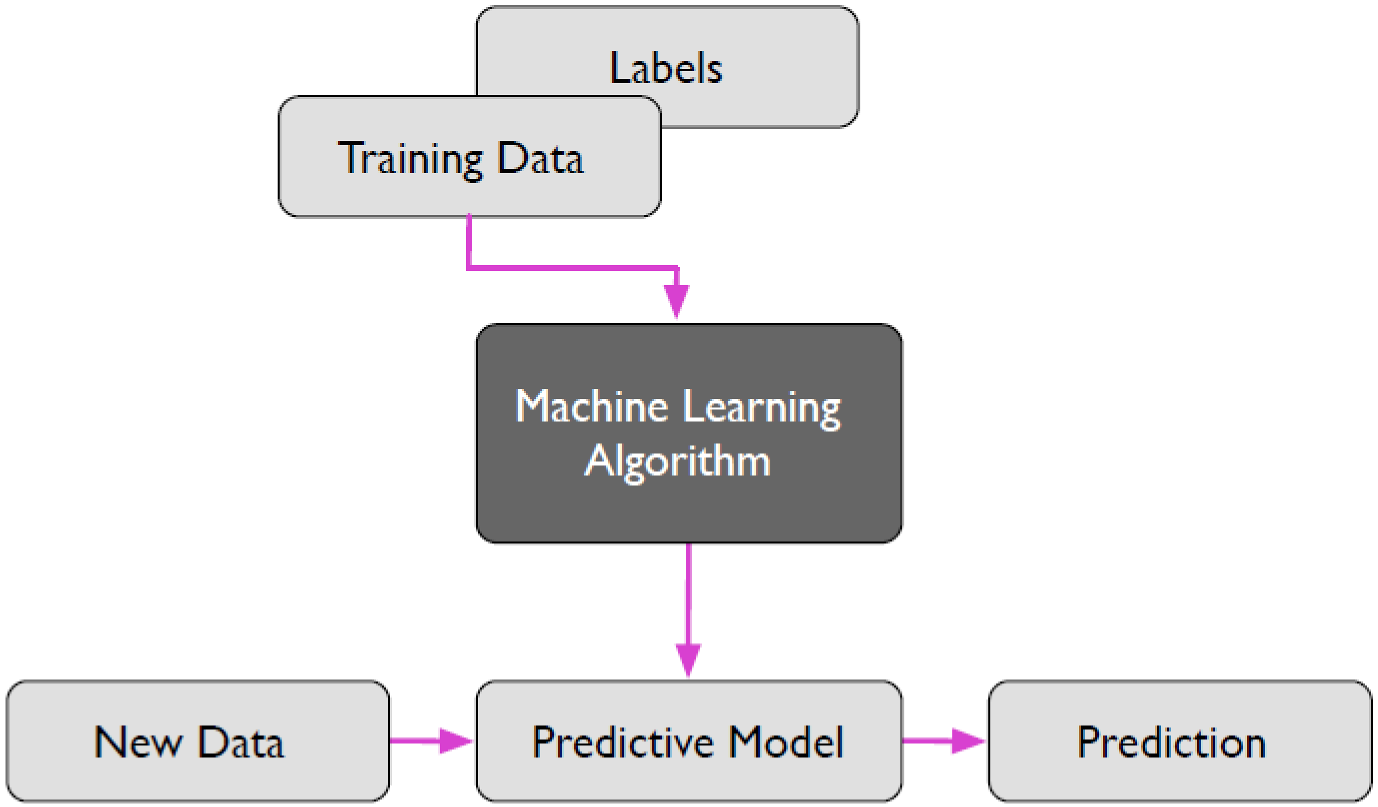
ML Terminology

- **Training example**: A row in the table representing the dataset. Synonymous to an observation, training record, training instance, training sample.
- **Feature**: a column in the table representing the dataset. Synonymous to predictor, variable, input, attribute, covariate.
- **Targets**: What we want to predict. Synonymous to outcome, output, ground truth, response variable, dependent variable, (class) label (in classification).
- **Output / prediction**: use this to distinguish from targets; here, means output from the model.

Lecture 1 Overview

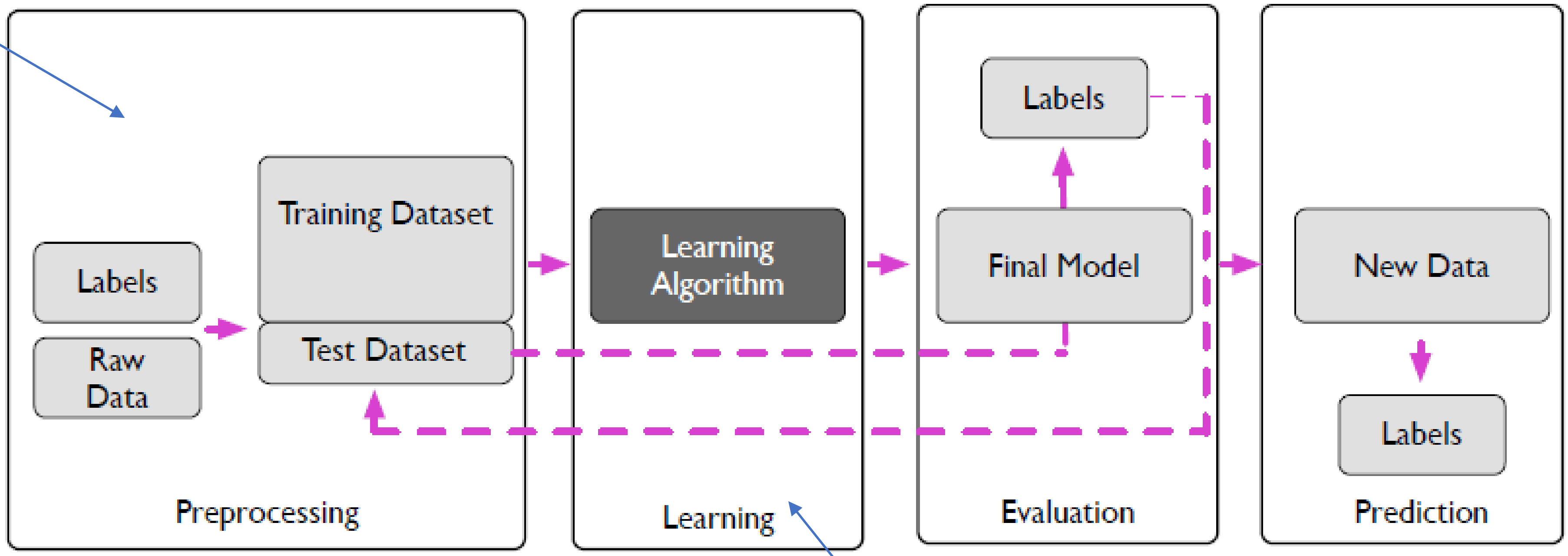
- What is machine learning ?
- Categories of machine learning
- Notation
- Approaching a machine learning application

Supervised Learning Workflow -- Overview



The framework

Feature Extraction and Scaling
Feature Selection
Dimensionality Reduction
Sampling



Model Selection
Cross-Validation
Performance Metrics
Hyperparameter Optimization

The framework

1. Define the problem to be solved.
2. Collect (labeled) data.
3. Choose an algorithm class.
4. Choose an optimization metric or measure for learning the model.
5. Choose a metric or measure for evaluating the model.

Evaluation -- Misclassification Error

$$L(\hat{y}, y) = \begin{cases} 0 \; if \, \hat{y} = y \\ 1 \; if \, \hat{y} \neq y \end{cases}$$

$$ERR_{\mathcal{D}_{\boldsymbol{test}}} = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}^{i}, y^{i})$$

ML Terminology

- **Loss function:** Often used synonymously with cost function; sometimes also called error function. In some contexts the loss for a single data point, whereas the cost function refers to the overall (average or summed) loss over the entire dataset. Sometimes also called empirical risk.