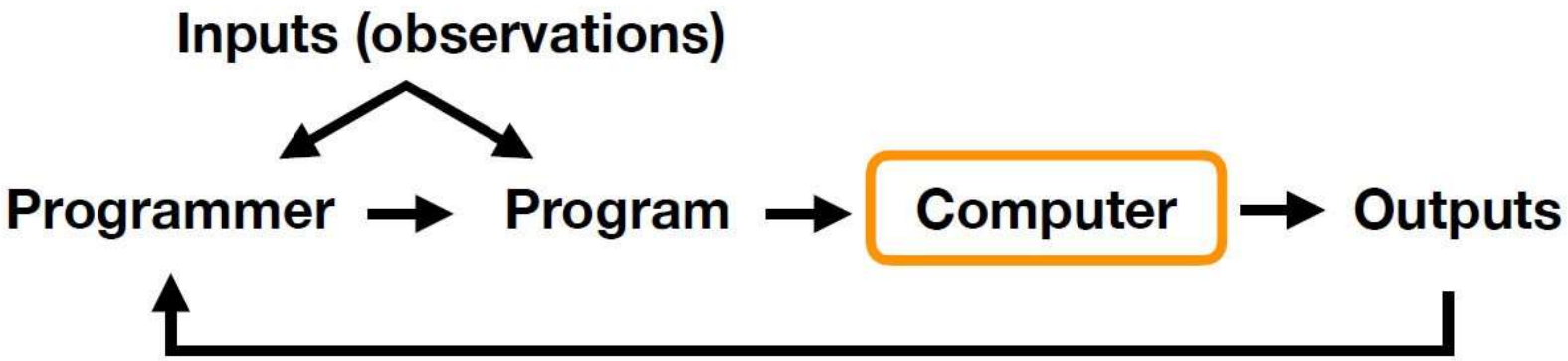
Introduction

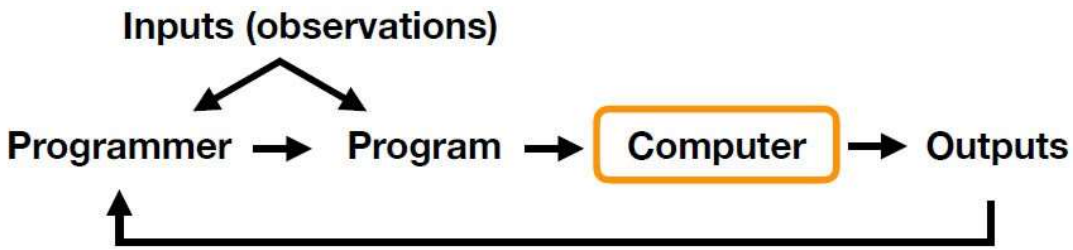
Fatemeh Mansoori

This slide are based on the slides by Sebastian Raschka for intro. to machine learning course

What is machine learning

The traditional Programming Paradigm





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

— *Arthur Samuel (1959)*

"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

Tom M Mitchell et al. "Machine learning. 1997". In: *Burr Ridge, IL: McGraw Hill* 45.37 (1997), pp. 870–877.

"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:

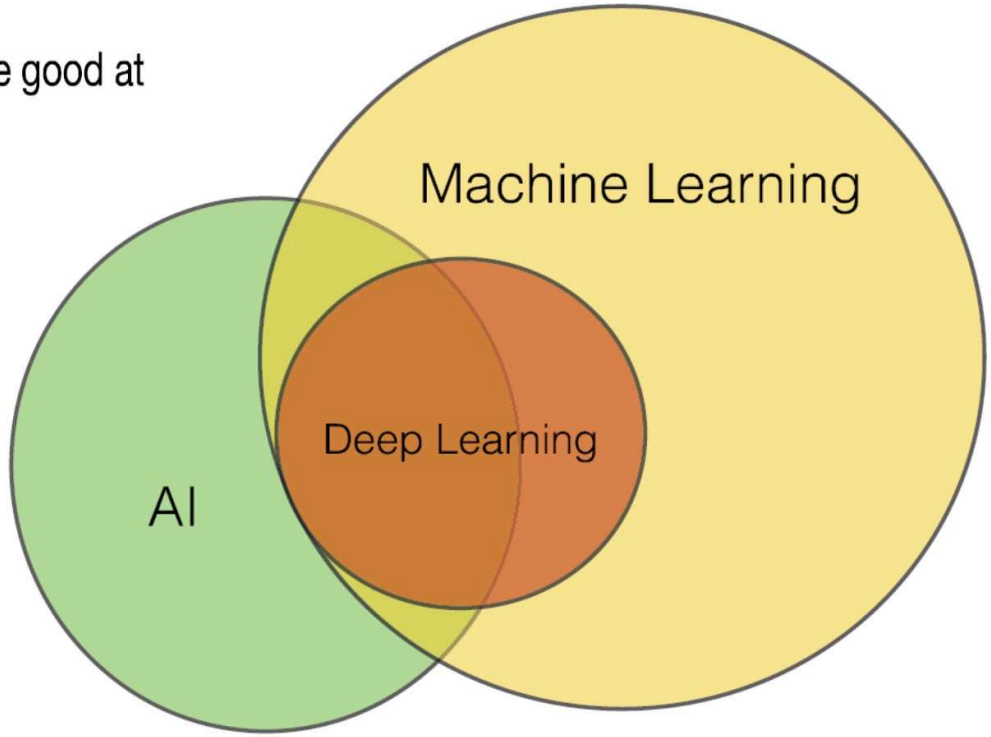
- Task T: ?
- Performance measure P : ?
- Training experience E: ?

The Connection between fields

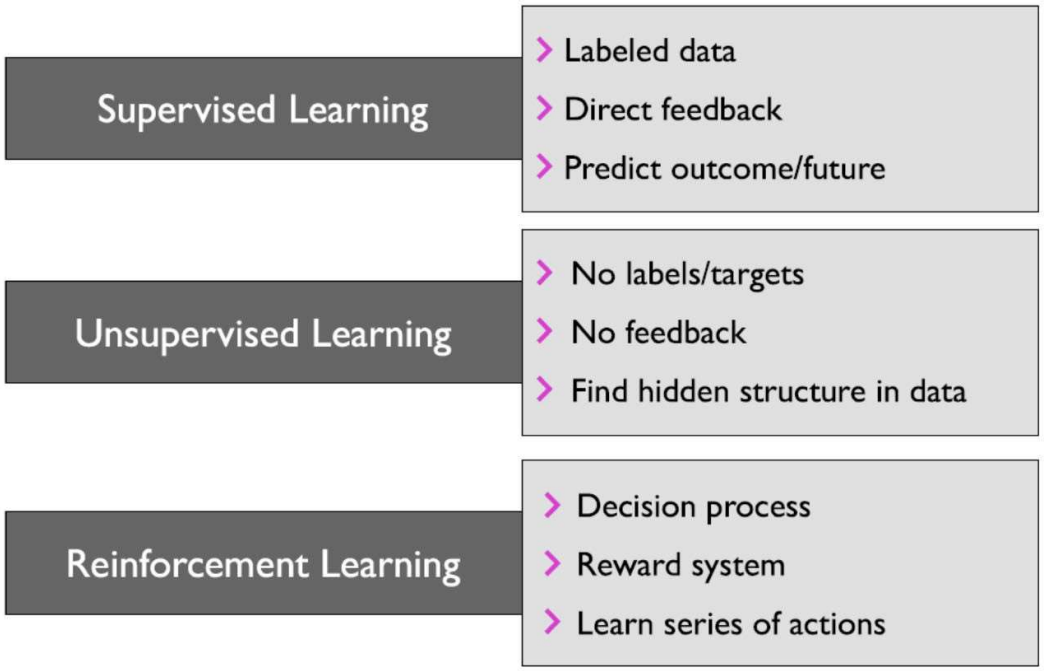
Artificial Intelligence (AI):
orig. subfield of computer science, solving tasks humans are good at

Narrow AI:
solving a particular task (playing a game, driving a car, ...)

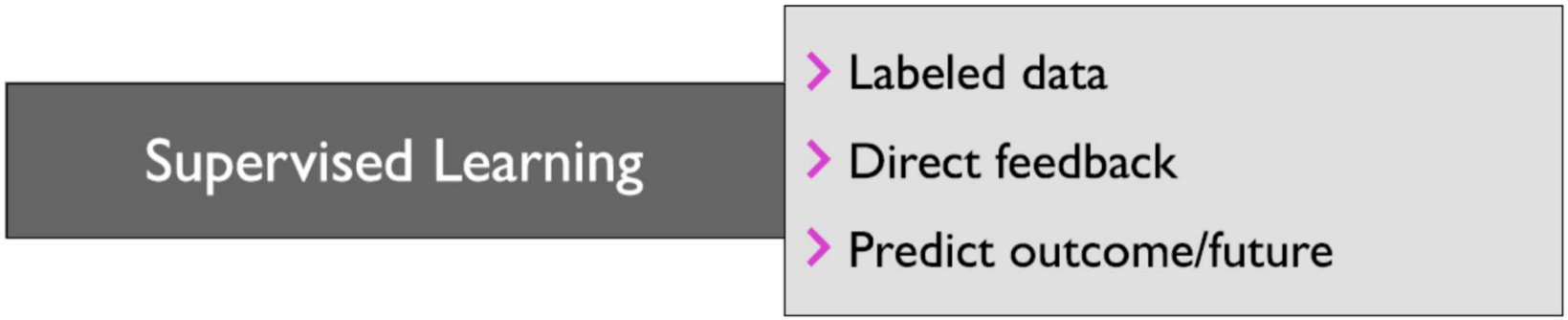
Artificial General Intelligence (AGI):
multi-purpose AI mimicking human intelligence across tasks



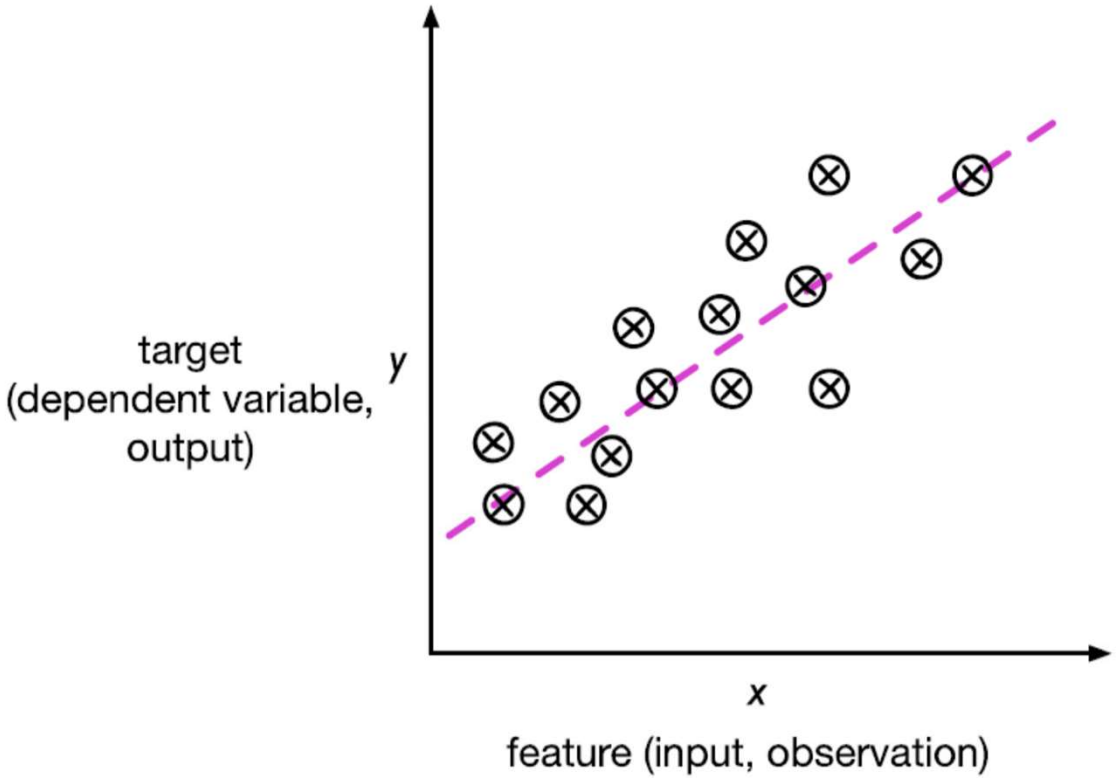
Categories of Machine Learning



Supervised Learning Is The Largest Subcategory



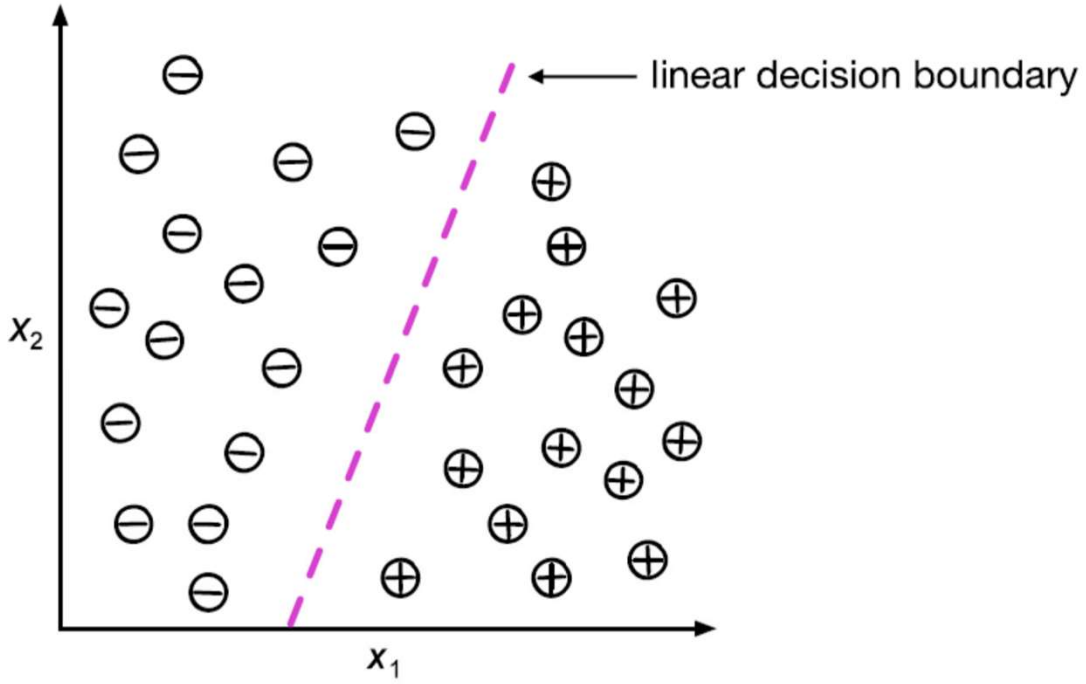
Supervised Learning 1: Regression



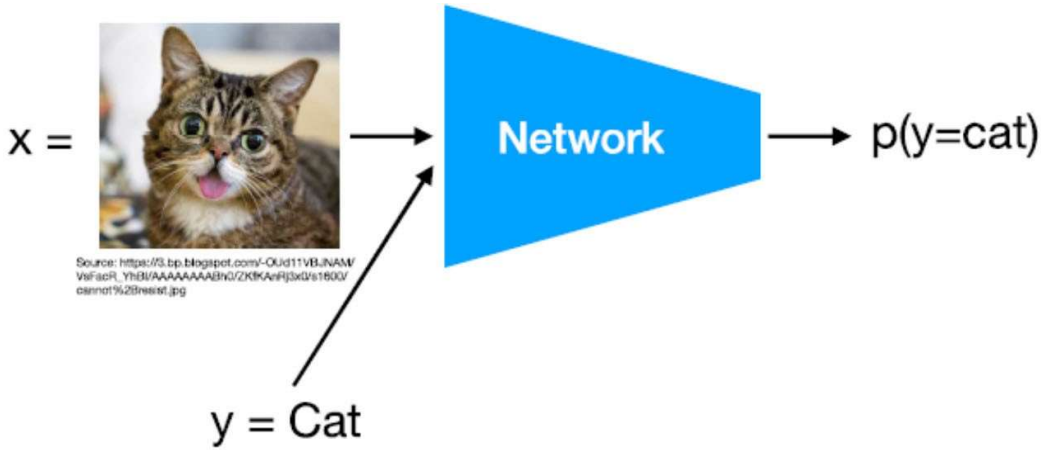
Supervised Learning 2: Classification

Binary classification example with two *features* ("independent" variables, predictors)

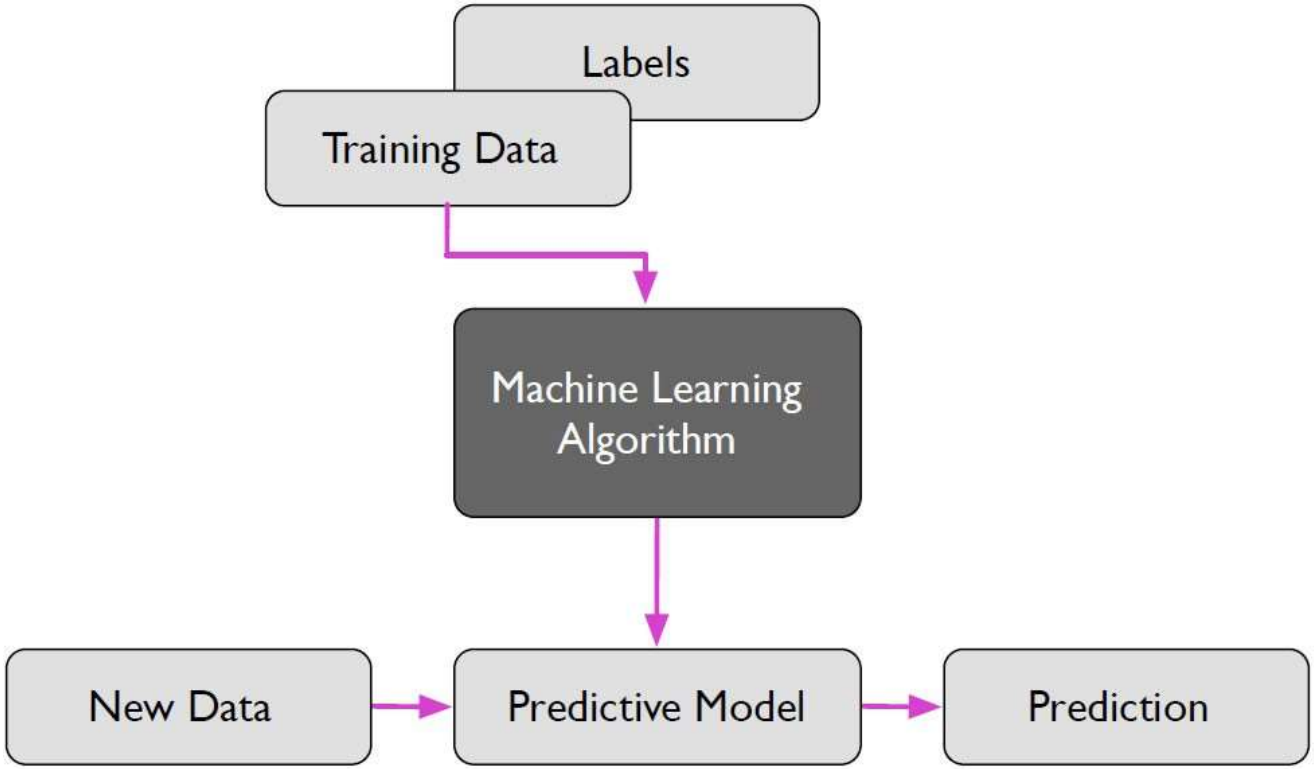
What are the class labels (y's)?



Classification works like this

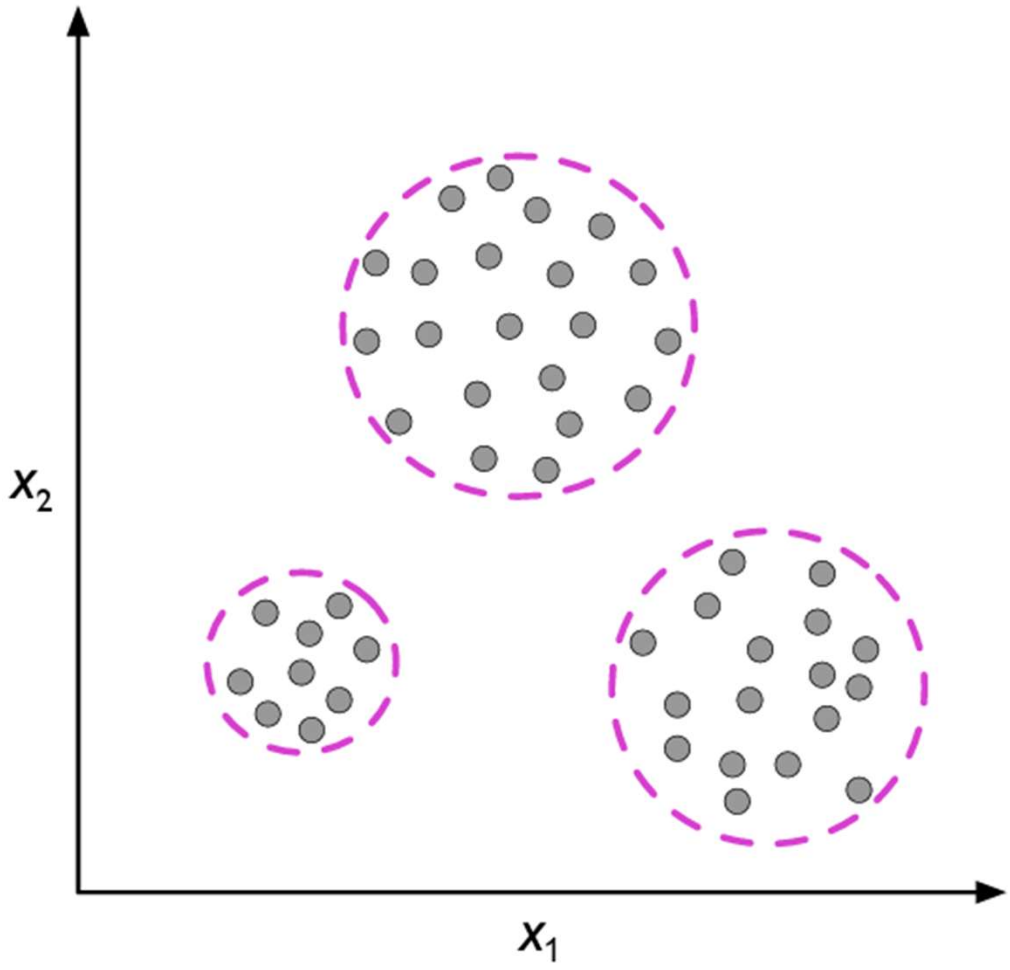


Supervised Learning Workflow -- Overview



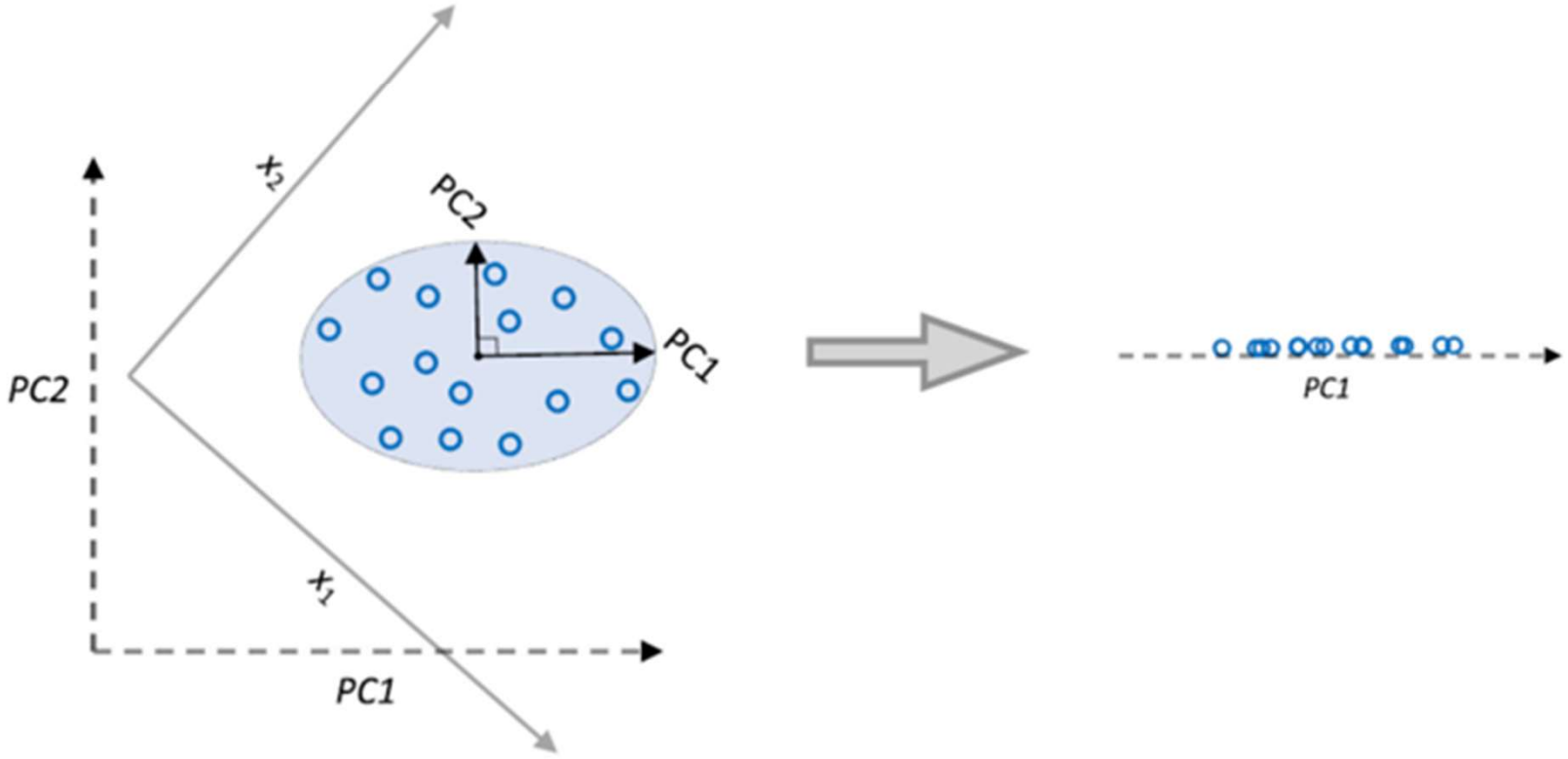
The 2nd Subcategory Of ML (And DL)

Unsupervised Learning -- Clustering

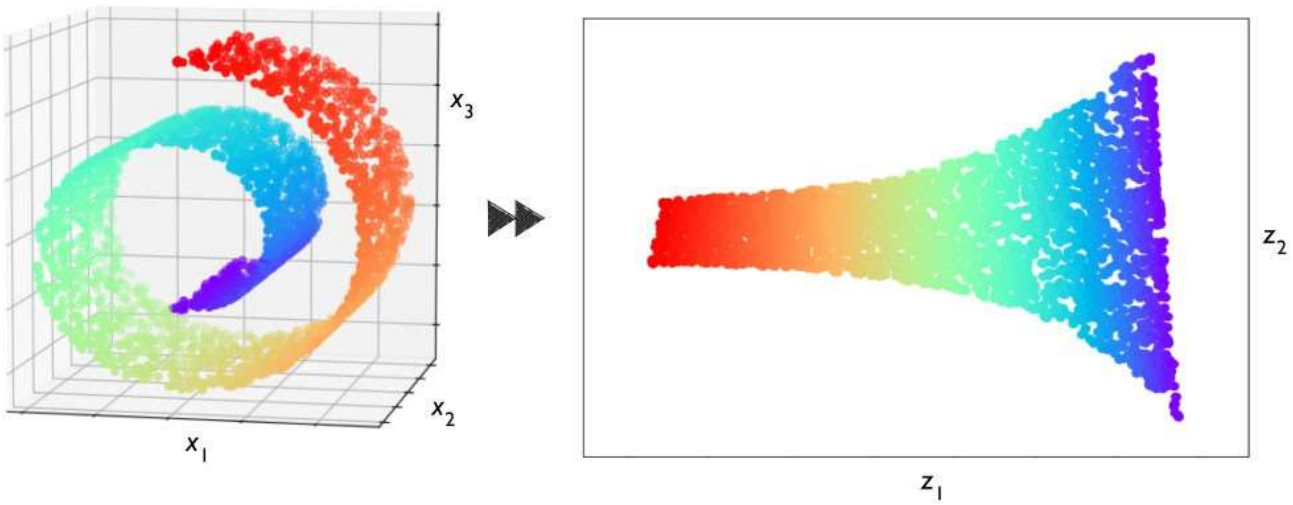


Representation Learning/Dimensionality Reduction

E.g., Principal Component Analysis (PCA)

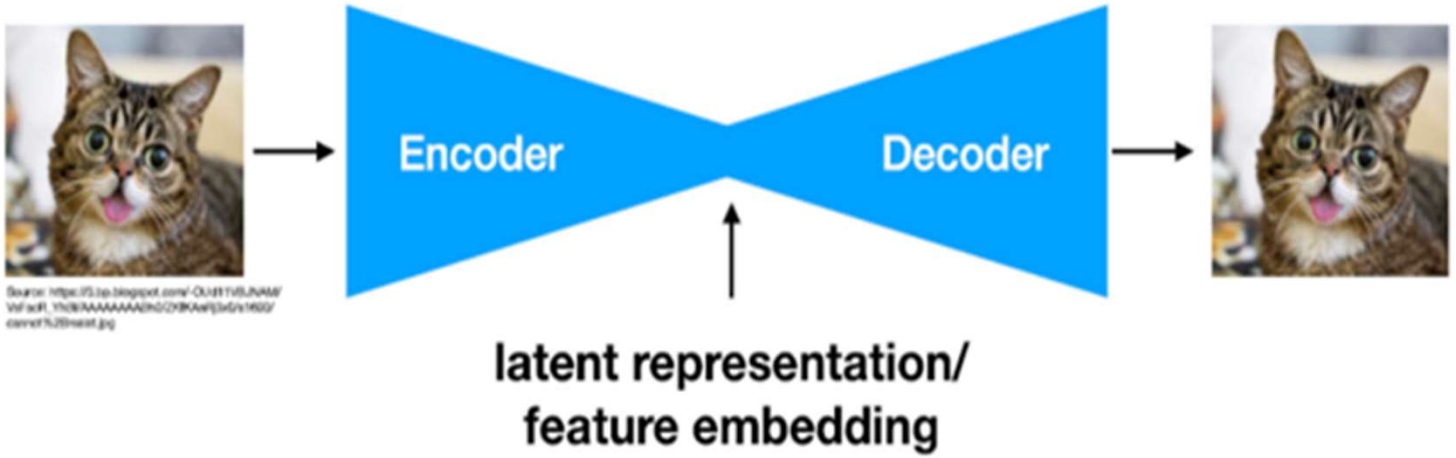


Unsupervised Learning--Dimensionality Reduction

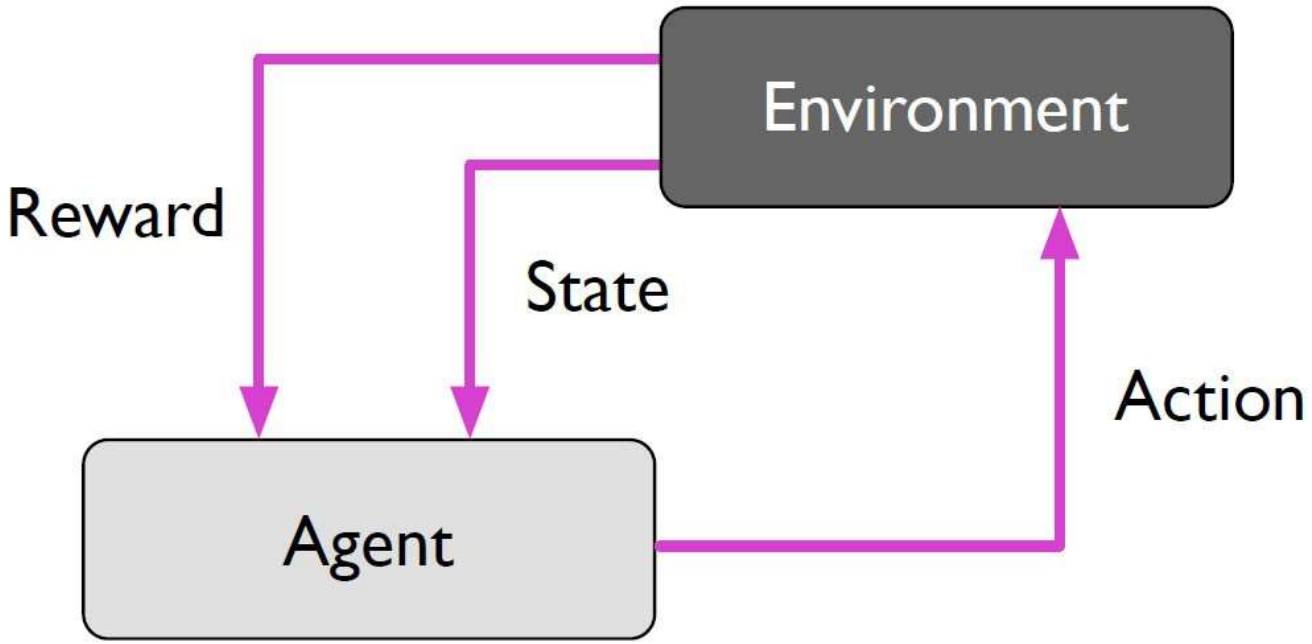


Representation Learning/Dimensionality Reduction

E.g., Autoencoders



Reinforcement Learning



Semi-Supervised Learning

- mix between supervised and unsupervised learning
- some training examples contain outputs, but some do not
- use the labeled training subset to label the unlabeled portion of the training set, which we then also utilize for model training

Semi-Supervised Learning

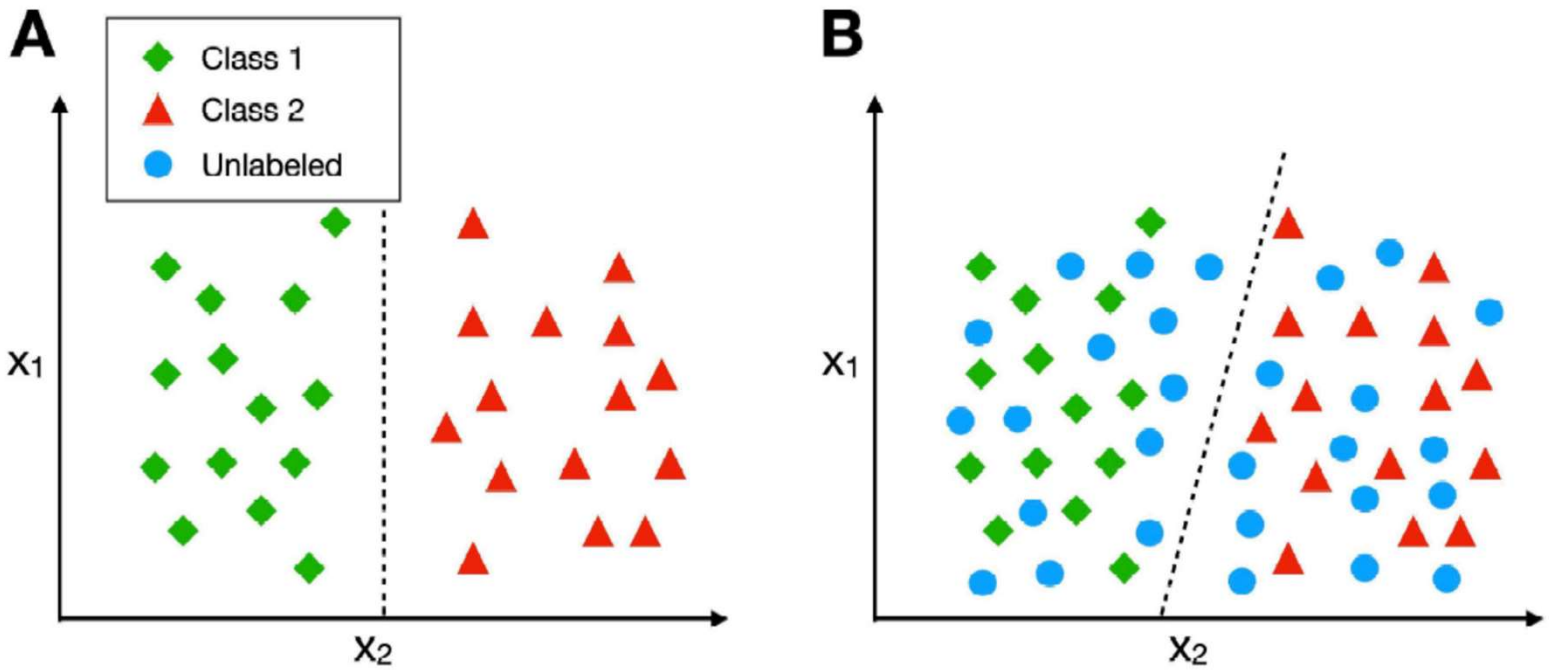


Illustration of semi-supervised learning incorporating unlabeled examples. (A) A decision boundary derived from the labeled training examples only. (B) A decision boundary based on both labeled and unlabeled examples.

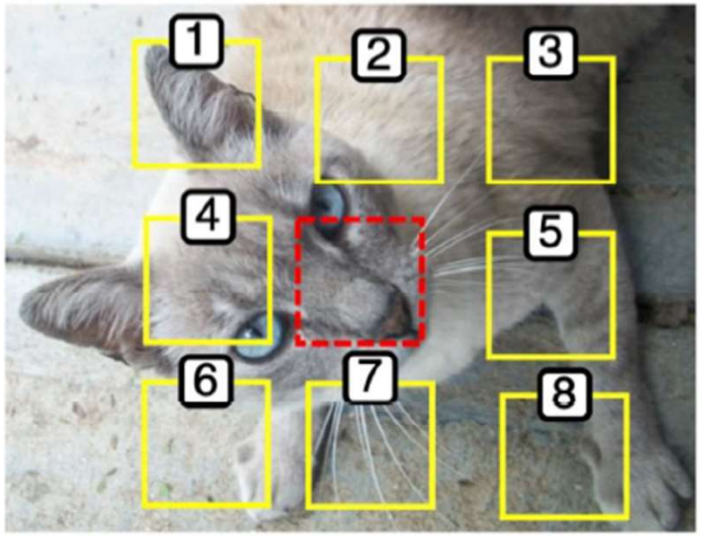
Self-supervised Models

- How do we use unlabeled data for learning representations?
  - Predict next word / patch of image
  - Predict missing word / patch of image
  - Predict if two images are related (contrastive learning)

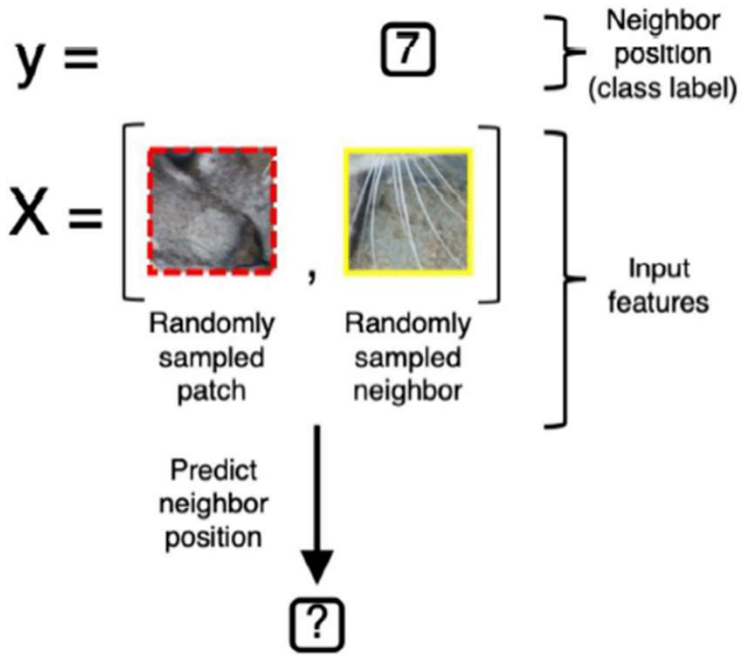
- Making poweful foundation models using this approach

Self-Supervised Learning

A



B



Self-supervised learning via context prediction. (A) A random patch is sampled (red square) along with 9 neighboring patches. (B) Given the random patch and a random neighbor patch, the task is to predict the position of the neighboring patch relative to the center patch (red square).

Machine Learning vs Deep Learning

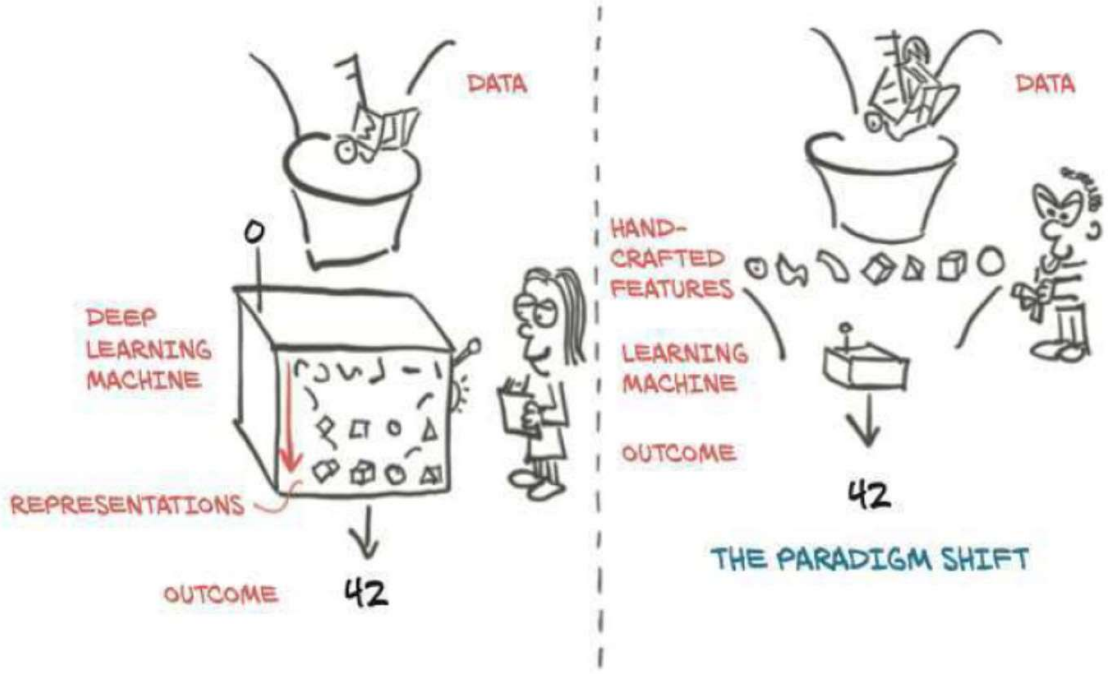


Image source: Stevens et al., *Deep Learning with PyTorch*. Manning, *2020*

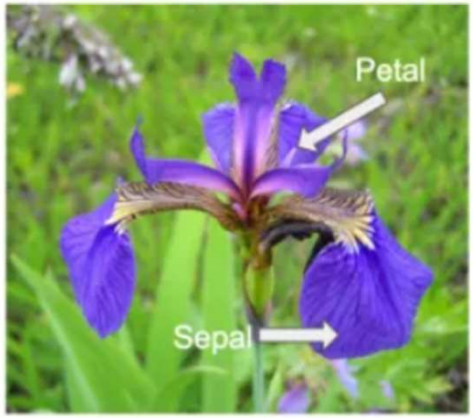
Structured vs Unstructured Data

Feature vector of the 1st training example

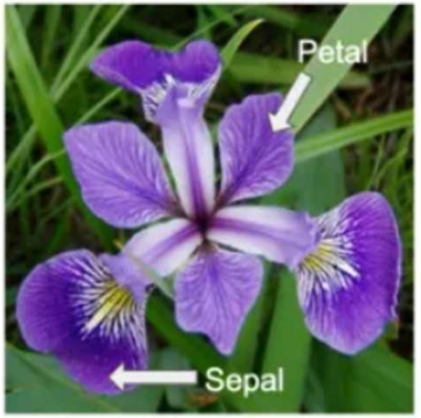
Class label

Index	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
...	...	...	...	...	...
150	5.9	3	5.1	1.8	Iris-virginica

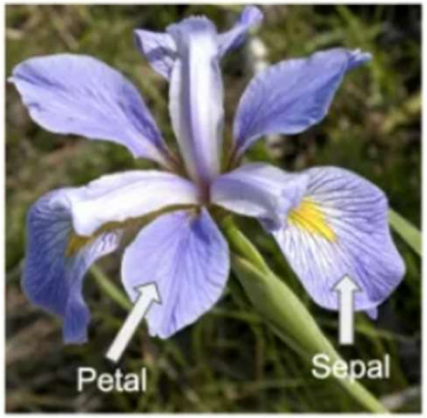
Iris setosa



Iris versicolor



Iris virginica



https://peaceadegbite1.medium.com/iris-flower-classification-60790e9718a1

Machine Learning vs Deep Learning

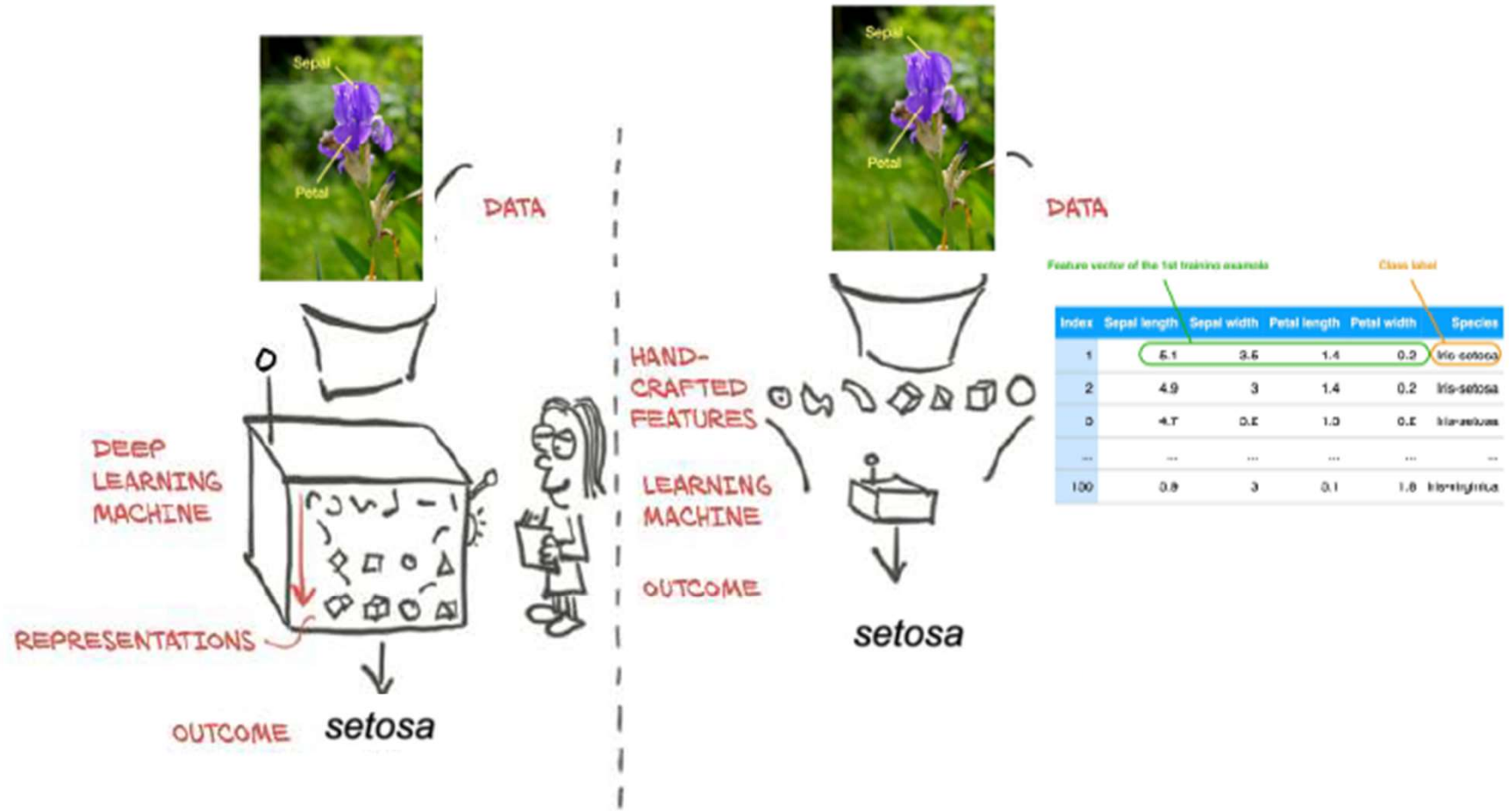
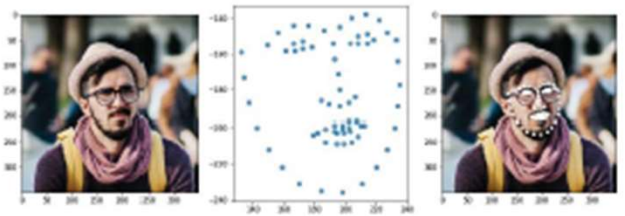


Image source: Stevens et al., *Deep Learning with PyTorch.* Manning, *2020*

- ***supervised learning:***
  learn function to map input *x* (features) to output *y* (targets)

- ***structured data:***
  databases, spreadsheets/csv files

- ***unstructured data:***
  features like image pixels, audio signals, text sentences
  (before DL, extensive feature engineering was required)



Source: http://rasbt.github.io/mlxtend/user_guide/image/extract_face_landmarks/

Supervised Learning (More Formal Notation)

"training examples"

Training set: $\mathscr{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \ldots, n\},$

Unknown function: $f(\mathbf{x}) = y$

Hypothesis: $h(\mathbf{x}) = \hat{y}$

sometimes *t* or *o*

Classification

Regression

$h : \mathbb{R}^m \to \mathscr{Y}, \quad \mathscr{Y} = \{1, \ldots, k\}$

$h : \mathbb{R}^m \to \mathbb{R}$

Learning Model

- The **Learning Model** consists of:
  - **Hypothesis Set**: Defines the possible functions $\mathcal{H} = \{h(x,\theta)|\theta \in \Theta\}$, where $h(x,\theta)$ represents candidate functions and θ is the learning parameters of problem.
  - **Learning Algorithm**: Find $\theta^* \in \Theta$ such that $h(x,\theta^*) \approx f(x)$.
- Both work together to map inputs x to outputs y with minimized error.
- In other words, θ^* is best parameters to predict outputs using chosen hypothesis.

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix}$$

Feature vector

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y^{[1]} \\ y^{[2]} \\ \vdots \\ y^{[n]} \end{bmatrix}$$

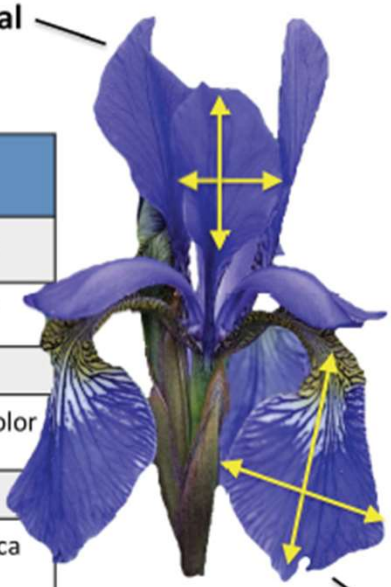
Input features

Data Representation (structured data)

$m =$ ____

$n =$ ____

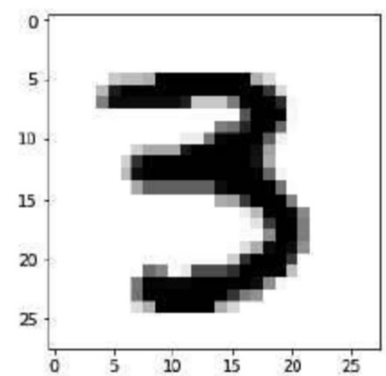
	Sepal length	Sepal width	Petal length	Petal width	
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



Data Representation (unstructured data; images)

"traditional methods"

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.20392157, 0.2627451, 0.2627451, 0.31764707, 0.99607843, 0.99607843, 0.99607843
, 0.99607843, 1.0, 0.99607843, 0.99607843, 0.99607843, 0.30980393, 0.14509805, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.27450982, 0.8862745, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.9
9215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.7764706, 0.15294118, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5019608, 0.9529412, 0.9529412, 0.9529412, 0.9529412, 0.9529412,
0.9529412, 0.87058824, 0.21568628, 0.21568628, 0.21568628, 0.5176471, 0.98039216, 0.99215686, 0.99215686, 0.8392157
, 0.023529412, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.6627451, 0.99215686, 0.99215686, 0.99215686, 0.03137255, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.49803922, 0.5529412, 0.84705883, 0.99215686, 0.99
215686, 0.59607846, 0.015686275, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.06666667, 0.07450981, 0.5411765, 0.972549, 0.99215686, 0.99215686, 0.99215686, 0.627451, 0.05490196, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1882353, 0.33333334, 0.33333334, 0.33333334
, 0.8980392, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.92941177, 0.3647059, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.18039216, 0.8156863, 0.99215686, 0.99215686, 0.99215686, 0
.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.91764706, 0.32941177, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.26666668, 0.9411765, 0.99215686, 0.99215686, 0.99215686, 0
.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.59607846, 0.03529412, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.34901962, 0.61960787, 0.61960787, 0.61960787,
0.61960787, 0.61960787, 0.61960787, 0.9529412, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.54901963, 0.011764
706, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.325
4902, 0.3647059, 0.78431374, 0.99215686, 0.99215686, 0.99215686, 0.5019608, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.07058824, 0.27058825, 0.99215686, 0.99
215686, 0.96862745, 0.45882353, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.10980392, 0.7647059, 0.99215686, 0.99215686, 0.50980395, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.44705883, 0.992156
86, 0.99215686, 0.50980395, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.1254902, 0.3254902, 0.9529412, 0.99215686, 0.9490196, 0.41960785, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.18431373, 0.8117647, 0.99215686,
0.99215686, 0.99215686, 0.76862746, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.57
64706, 0.47058824, 0.0, 0.08235294, 0.7019608, 0.7019608, 0.7019608, 0.81960785, 0.99215686, 0.99215686, 0.99215686
, 0.99215686, 0.24705882, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5372549, 0.984313
7, 0.98039216, 0.9607843, 0.9647059, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.9490196, 0.43921
57, 0.03137255, 0.003921569, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5529412, 0.992
15686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.7921569, 0.50980393, 0.41960785, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.14117648, 0.30980393, 0.99215
686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.32941177, 0.10901961, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]



Hypothesis Space Overview

- **Hypothesis (h)**: A mapping from input space $\mathcal{X}$ to output space $\mathcal{Y}$.
- **Linear Regression Hypothesis**:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_D x_D = \mathbf{w}^\top \mathbf{x}$$

- **Input Vector x**:

$$\mathbf{x} = [x_0 = 1, x_1, x_2, \ldots, x_D]$$

- **Parameter Vector w**:

$$\mathbf{w} = [w_0, w_1, w_2, \ldots, w_D]$$

Understanding Cost function

- In **hypothesis space**, we select a function $h(x; \mathbf{w})$ to approximate the true relationship between input x and output y.
- The objective is to minimize the difference between predicted values $h(x)$ and actual values y.
- This difference is quantified using **cost functions**, which guide us in choosing the optimal hypothesis.

What is Const Function ?

- A **cost function** measures how well the hypothesis $h(x;\mathbf{w})$ fits the training data.
- In regression problems, the most common error function is the **Squared Error (SE)**:

$$SE: \left(y^{(i)} - h(x^{(i)};\mathbf{w})\right)^2$$

- Cost function should measure all predictions. Thus a choice could be **Sum of Squared Errors (SSE)**:

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y^{(i)} - h(x^{(i)};\mathbf{w})\right)^2$$

- **Objective:** Minimize the cost function to find the best parameters **w**.

Sum of Squared Error

- **SSE** is widely used due to its simplicity and differentiability.
- Intuitively, it represents the squared distance between predicted and true values.
- Penalizes larger errors more severely than smaller ones (due to the square).
- For linear regression, it can be written as:

$$SSE = \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} \right)^2$$

Further Resources and Reading Materials

- "Introduction to Machine Learning and Deep Learning", article based on these slides https://sebastianraschka.com/blog/2020/intro-to-dl-ch01.html
- https://sebastianraschka.com/blog/2021/ml-course.html#l01---course-overview-introduction-to-machine-learning
- A. Sharifi Zarchi, “Introduction to Machine learning” course, Lecture slides.
- A. Ng and T. Ma, CS229 Lecture Notes.