Lecture 01

What Are Machine Learning And Deep Learning? An Overview.

STAT 453: Introduction to Deep Learning and Generative Models
Spring 2020

Sebastian Raschka
http://stat.wisc.edu/~sraschka/teaching/stat453-ss2020/

What You Will Learn Today

1/5 -- What Is Machine Learning?

2/5 -- The 3 Broad Categories of ML

3/5 -- Machine Learning Terminology and Notation

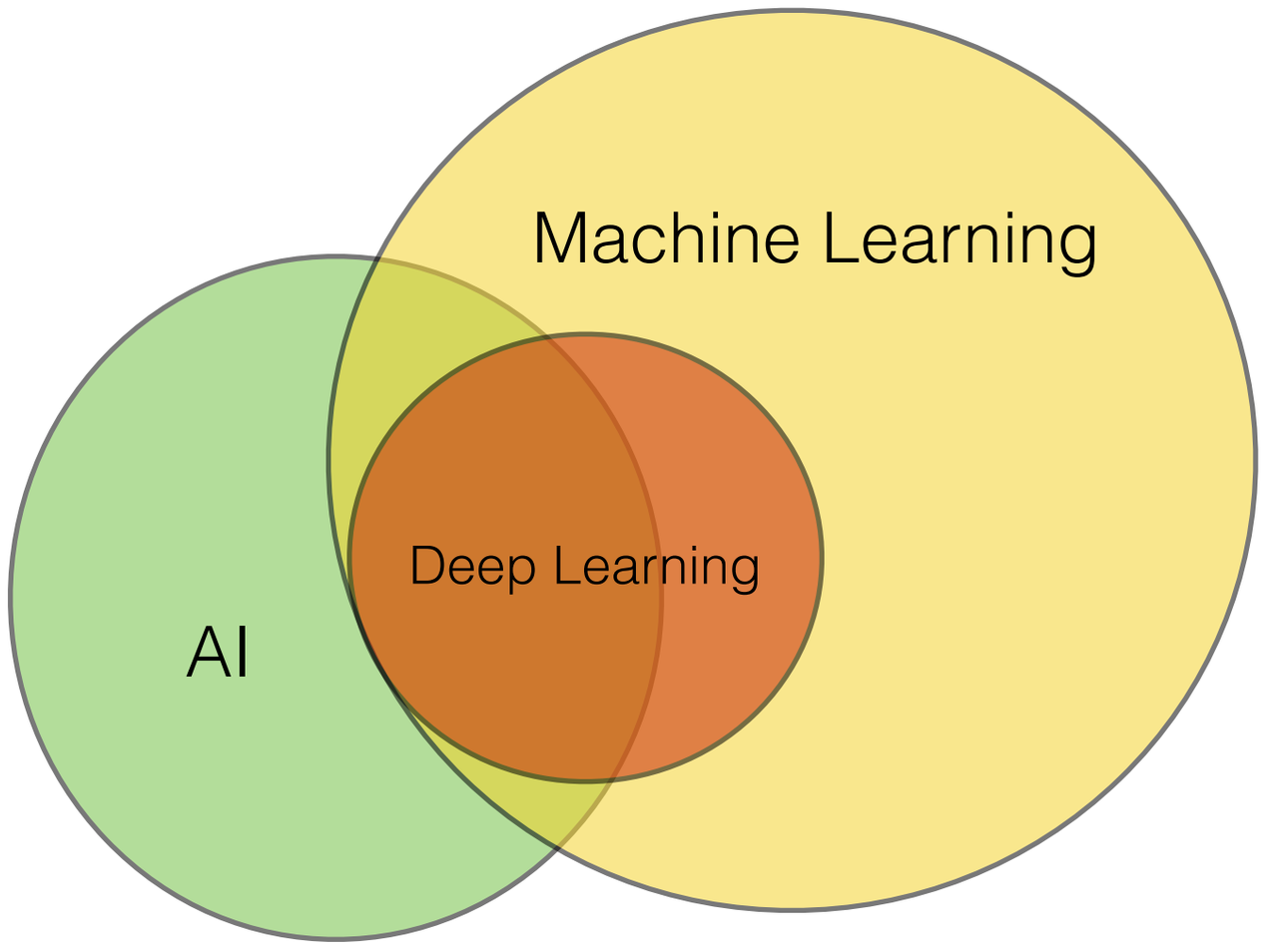
4/5 -- Machine Learning Modeling Pipeline

5/5 --The Practical Aspects: Our Tools!

What Is Machine Learning?

A short overview before we jump into Deep Learning

The Connection Between Fields

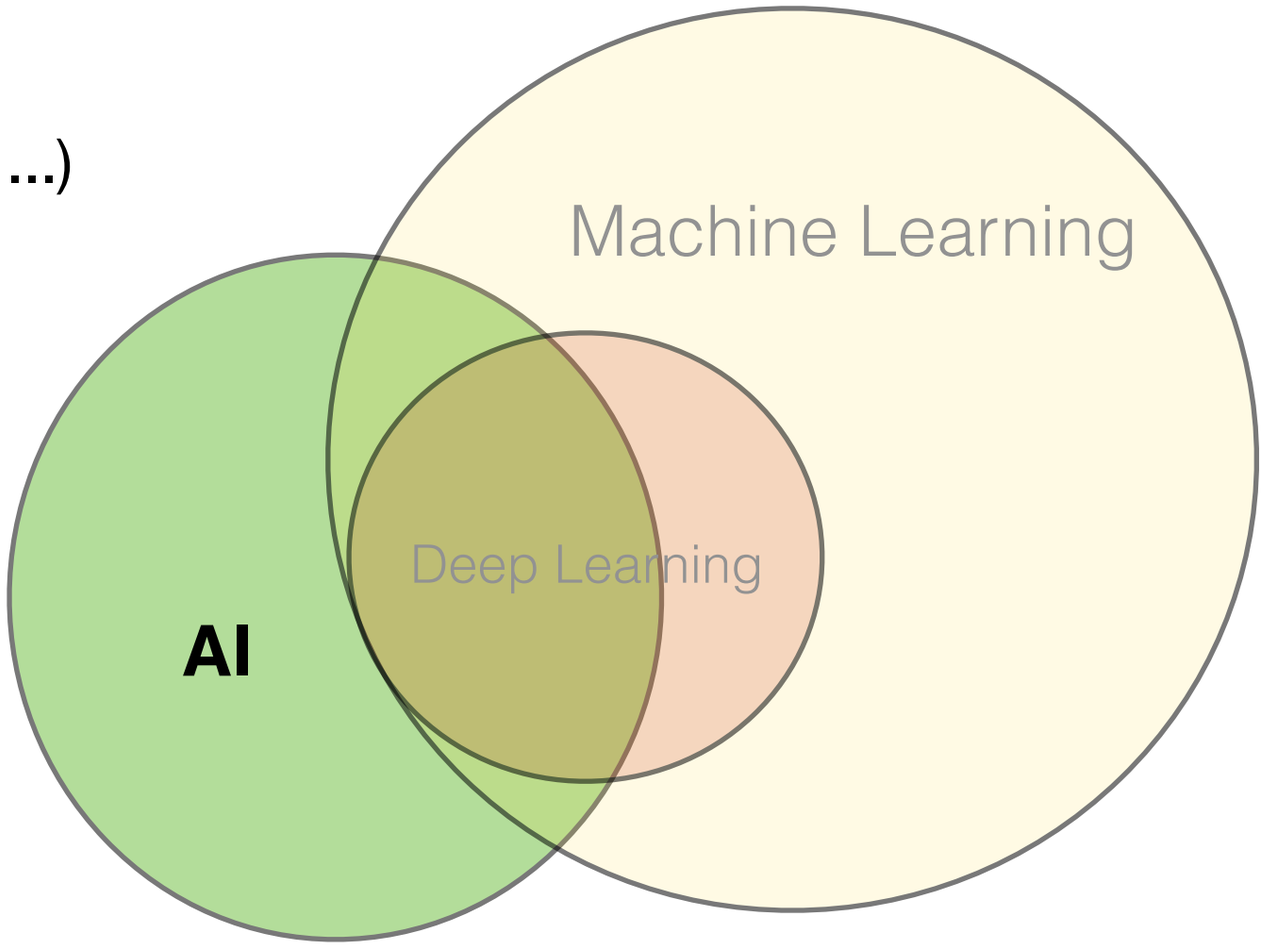


Different Types Of AI

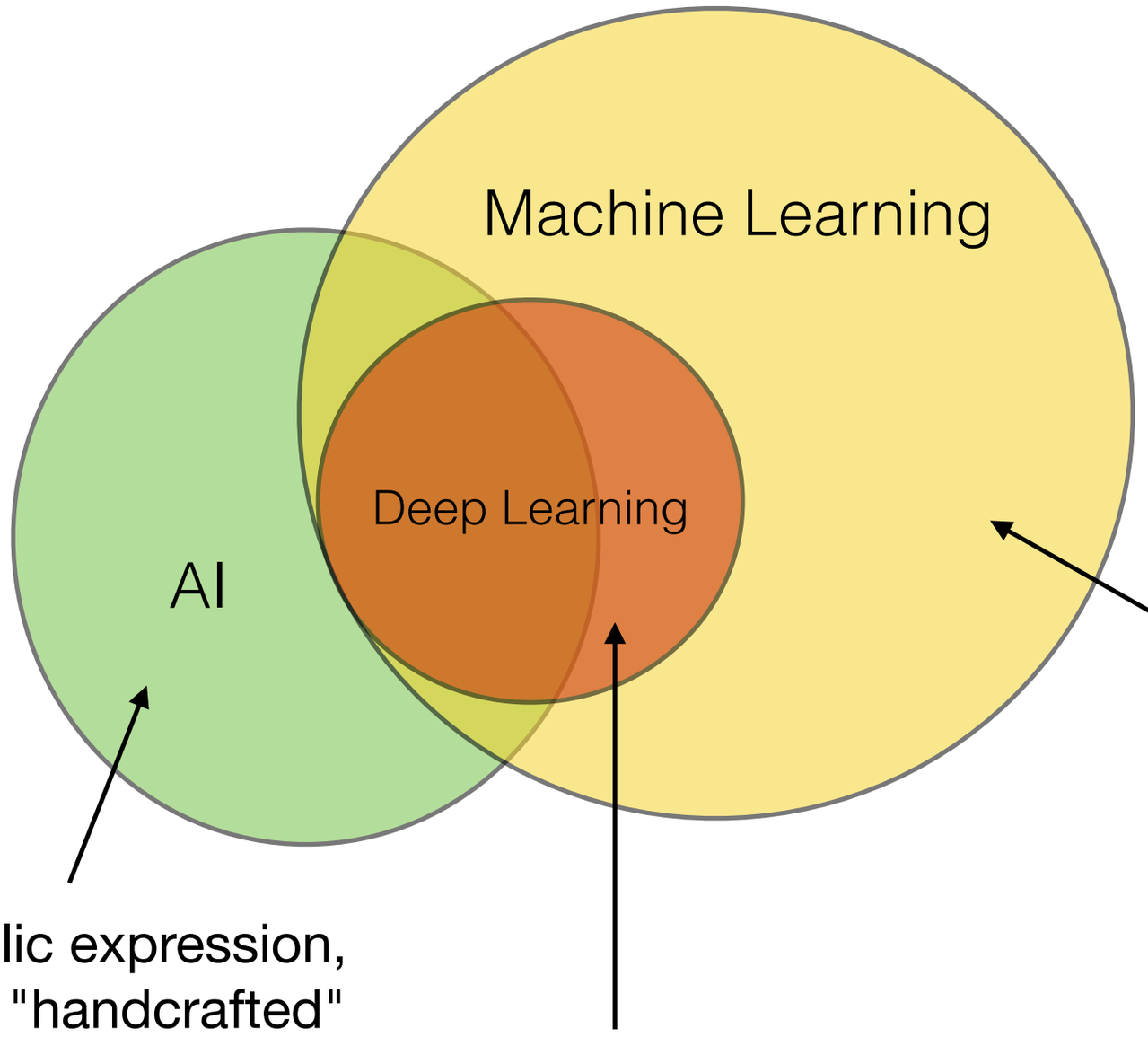
Artificial Intelligence (AI):
orig. subfield of computer science, solving tasks humans are good at (natural language, speech, image recognition, ...)

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

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



What This Course Is About

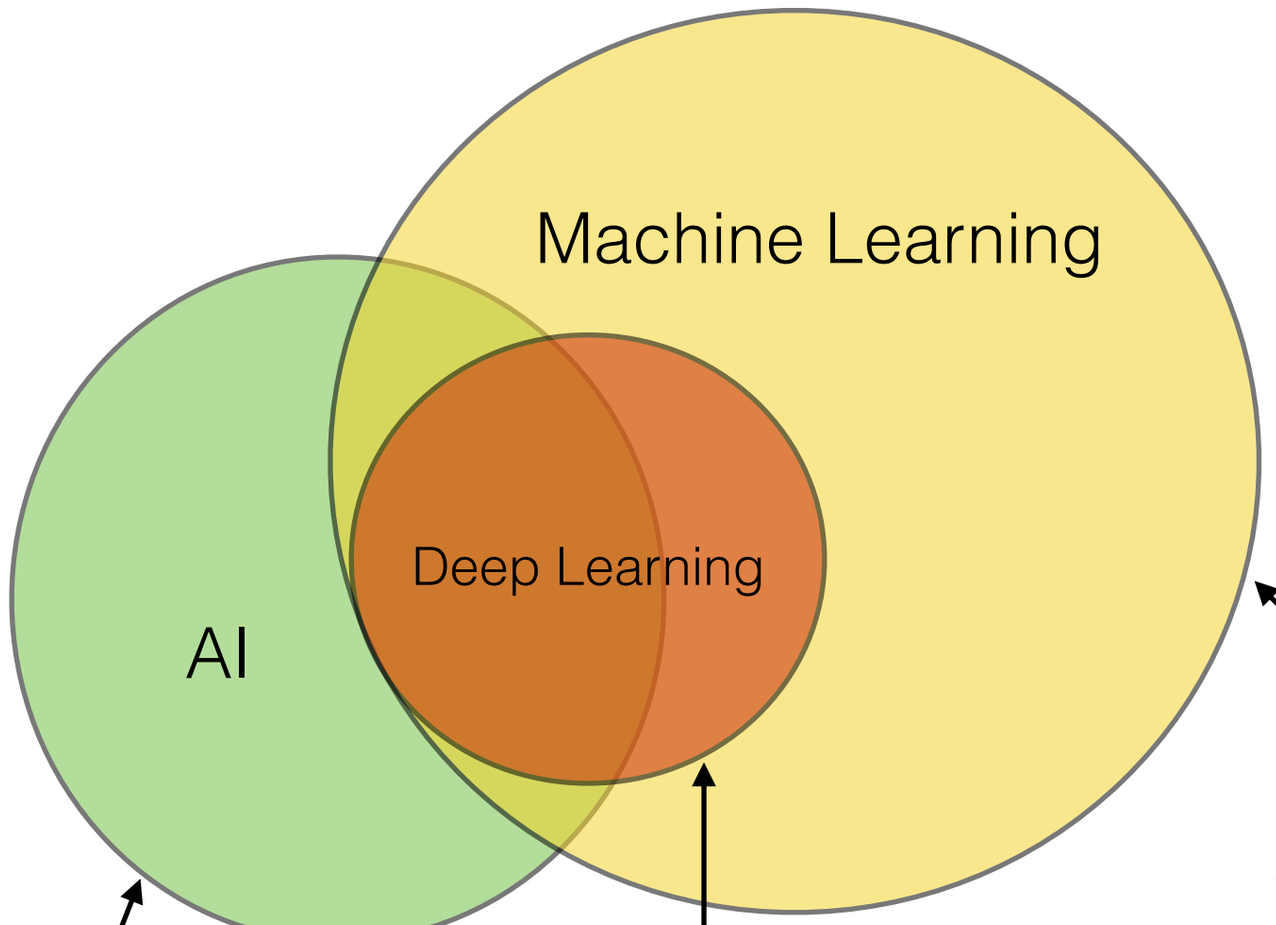


E.g., symbolic expression, logic rules / "handcrafted" nested if-else programming statements ...

Main focus of the course

E.g.,
generalized linear models,
tree-based methods,
"shallow" networks,
support vector machines,
nearest neighbors, ...

Examples From The Three Related "Areas"



= a non-biological system
that is intelligent
through rules

= algorithms that parameterize multi-layers
neural networks that then learn
representations of data with multiple layers
of abstraction

= algorithms that learn
models/representations/
rules automatically
from data/examples

What Is Machine Learning?

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

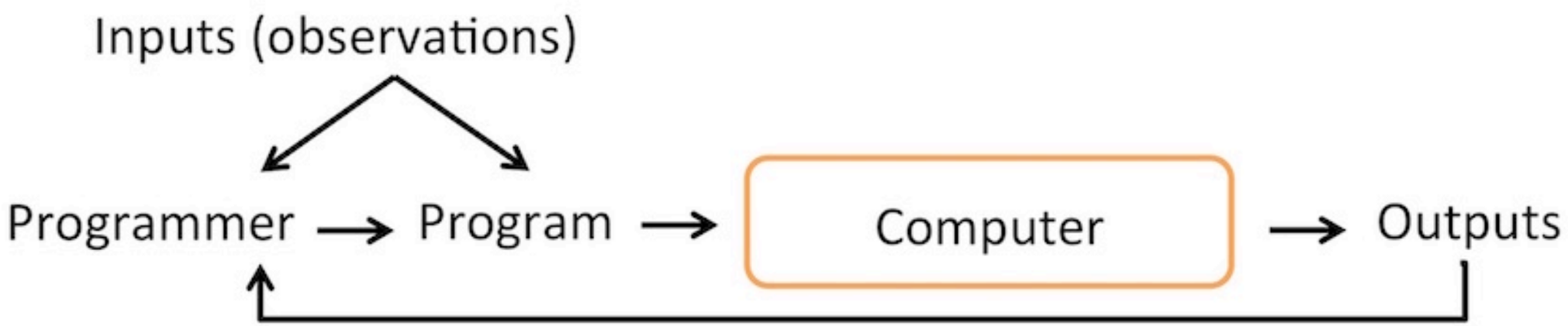
— Arthur L. Samuel, AI pioneer, 1959

[probably the first, and undoubtedly the most popular definition]

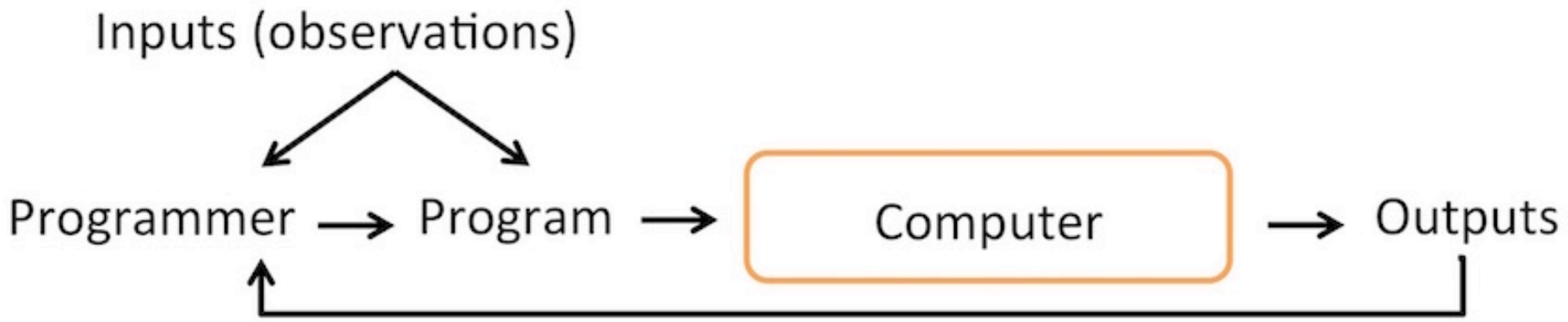
(This is likely not an original quote but a paraphrased version of Samuel's sentence "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.")

Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

The Traditional Programming Paradigm



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)

Machine Learning

Some Applications Of Machine Learning/Deep Learning

- Email spam detection
- Face detection and matching (e.g., iPhone X)
- Web search (e.g., DuckDuckGo, Bing, Google)
- Sports predictions
- Post office (e.g., sorting letters by zip codes)
- ATMs (e.g., reading checks)
- Credit card fraud
- Stock predictions

Some Applications Of Machine Learning/Deep Learning

- Smart assistants (Apple Siri, Amazon Alexa, ...)
- Product recommendations (e.g., Netflix, Amazon)
- Self-driving cars (e.g., Uber, Tesla)
- Language translation (Google translate)
- Sentiment analysis
- Drug design
- Medical diagnoses
- ...

The 3 Broad Categories of ML

(This also applies to DL)

1/5 -- What Is Machine Learning?
2/5 -- The 3 Broad Categories of ML
3/5 -- Machine Learning Terminology and Notation
4/5 -- Machine Learning Modeling Pipeline
5/5 --The Practical Aspects: Our Tools!

The 3 Broad Categories Of ML (And DL)

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

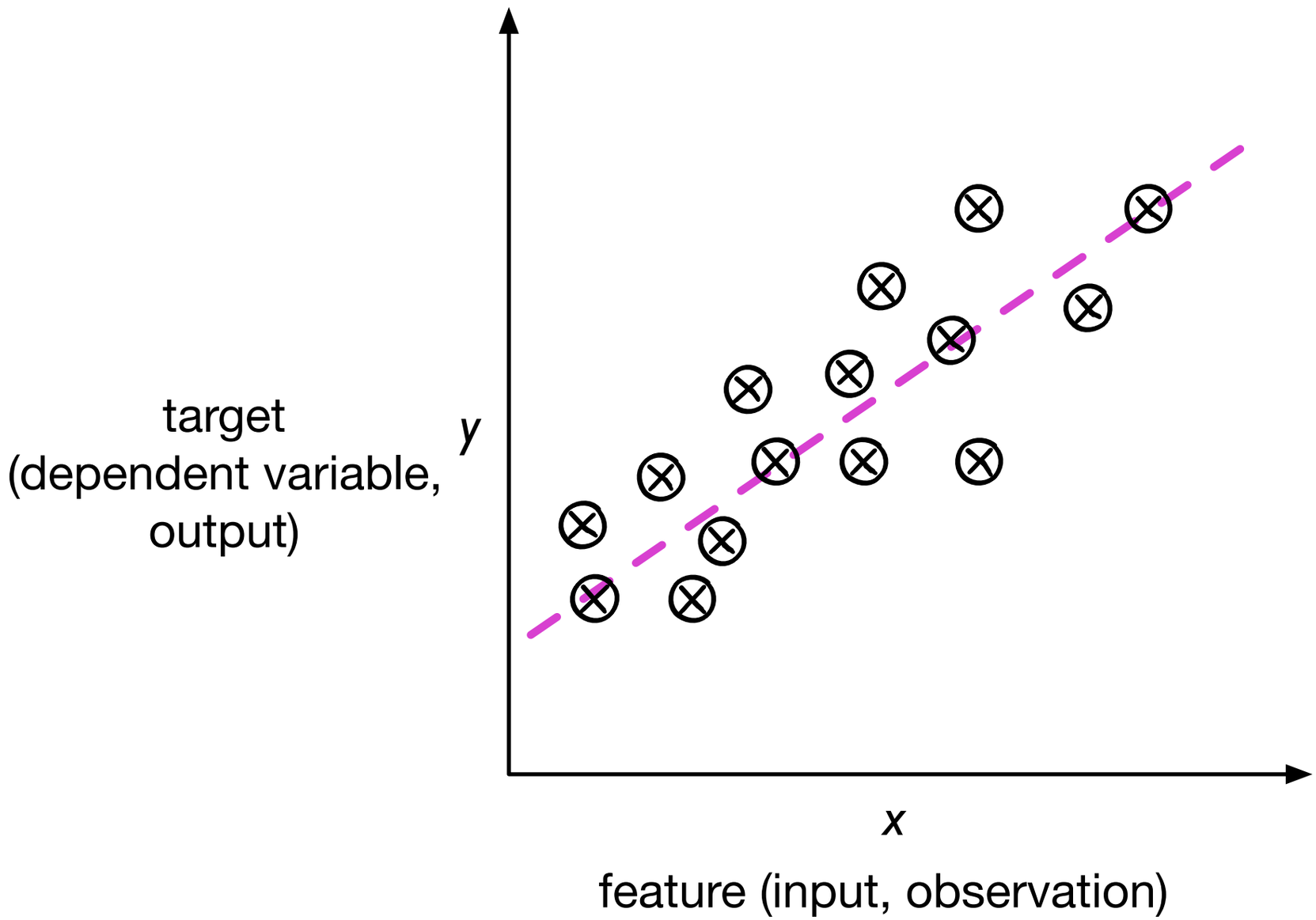
Supervised Learning Is The Largest Subcategory

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 1: Regression

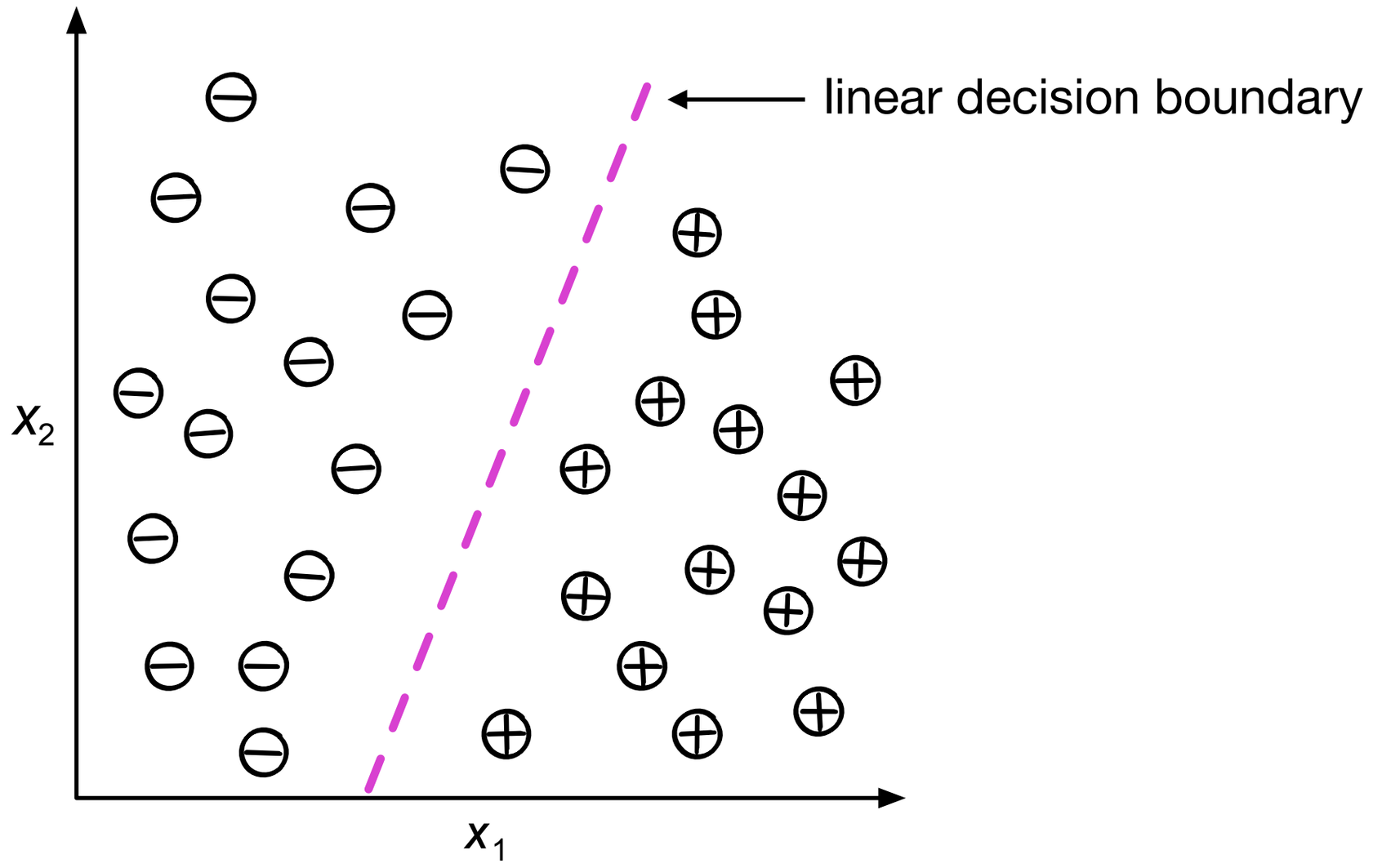


Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 2: Classification

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

What are the class labels (y's)?



Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 3: Ordinal regression

- Ordinal regression also called *ordinal classification* or *ranking* (although ranking is a bit different)

Order dependence like in metric regression,
but no metric distance

discrete values like in classification,
but order dependence

$$r_K \succ r_{K-1} \succ \ldots \succ r_1$$

E.g., movie ratings: *great* ≻ *good* ≻ *okay* ≻ *for genre fans* ≻ *bad*

Supervised Learning 3: Ordinal regression

- **Ranking:** Correct order matters
  (0 loss if order is correct, e.g., rank a collection of movies by "goodness")

- **Ordinal regression:** Correct label matters
  (E.g., age of a person in years; here, regard aging as a non-stationary process)

Excerpt from the UTKFace dataset
https://susanqq.github.io/UTKFace/

The 2n Subcategory Of ML (And DL)

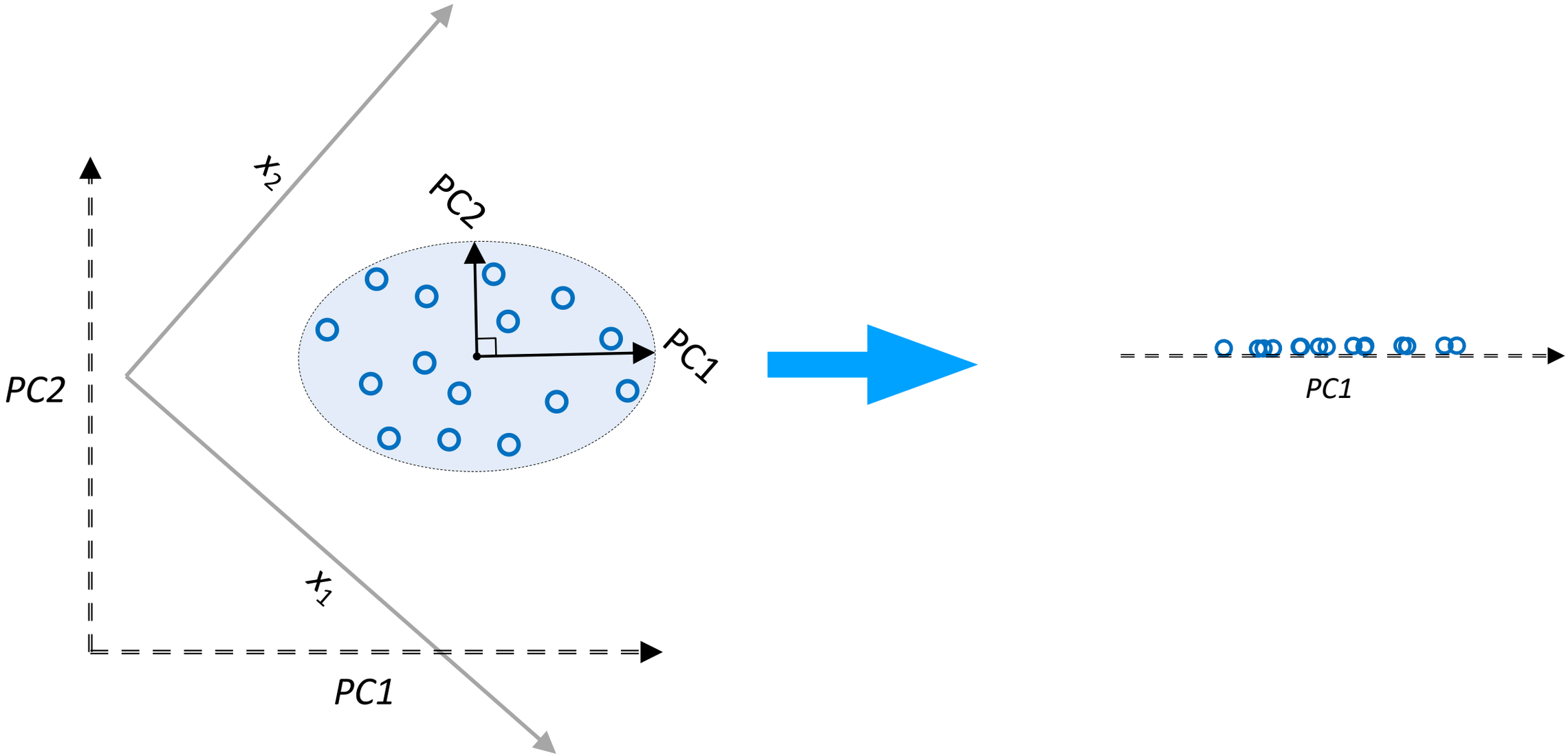
Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

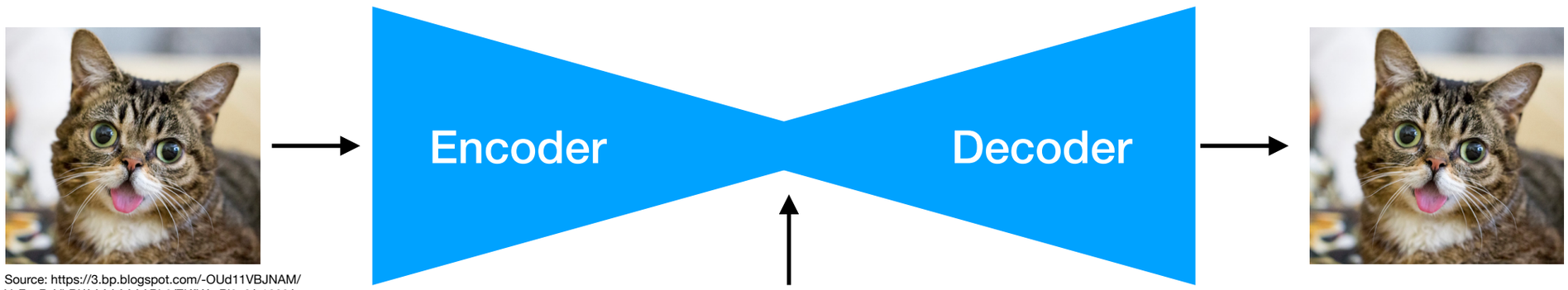
Unsupervised Learning 1: Representation Learning/Dimensionality Reduction

E.g., Principal Component Analysis (PCA)



Unsupervised Learning 1: Representation Learning/Dimensionality Reduction

E.g., Autoencoders

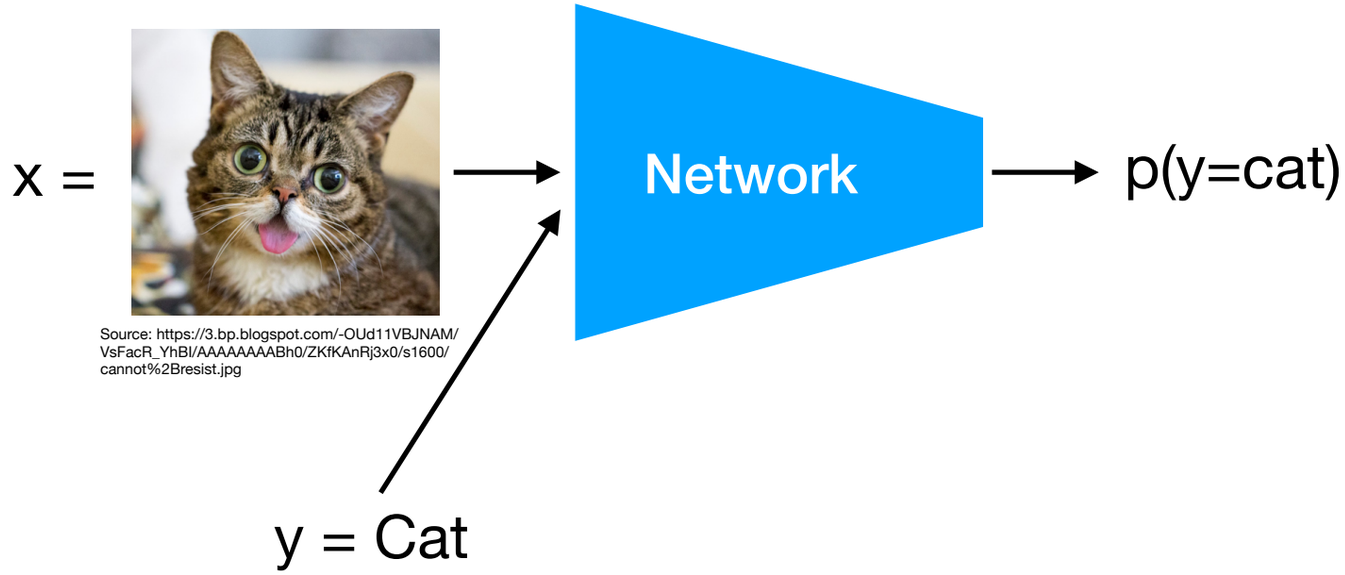


Source: https://3.bp.blogspot.com/-OUd11VBJNAM/VsFacR_YhBI/AAAAAAAABh0/ZKfKAnRj3x0/s1600/cannot%2Bresist.jpg

latent representation/ feature embedding

(covered later in this course)

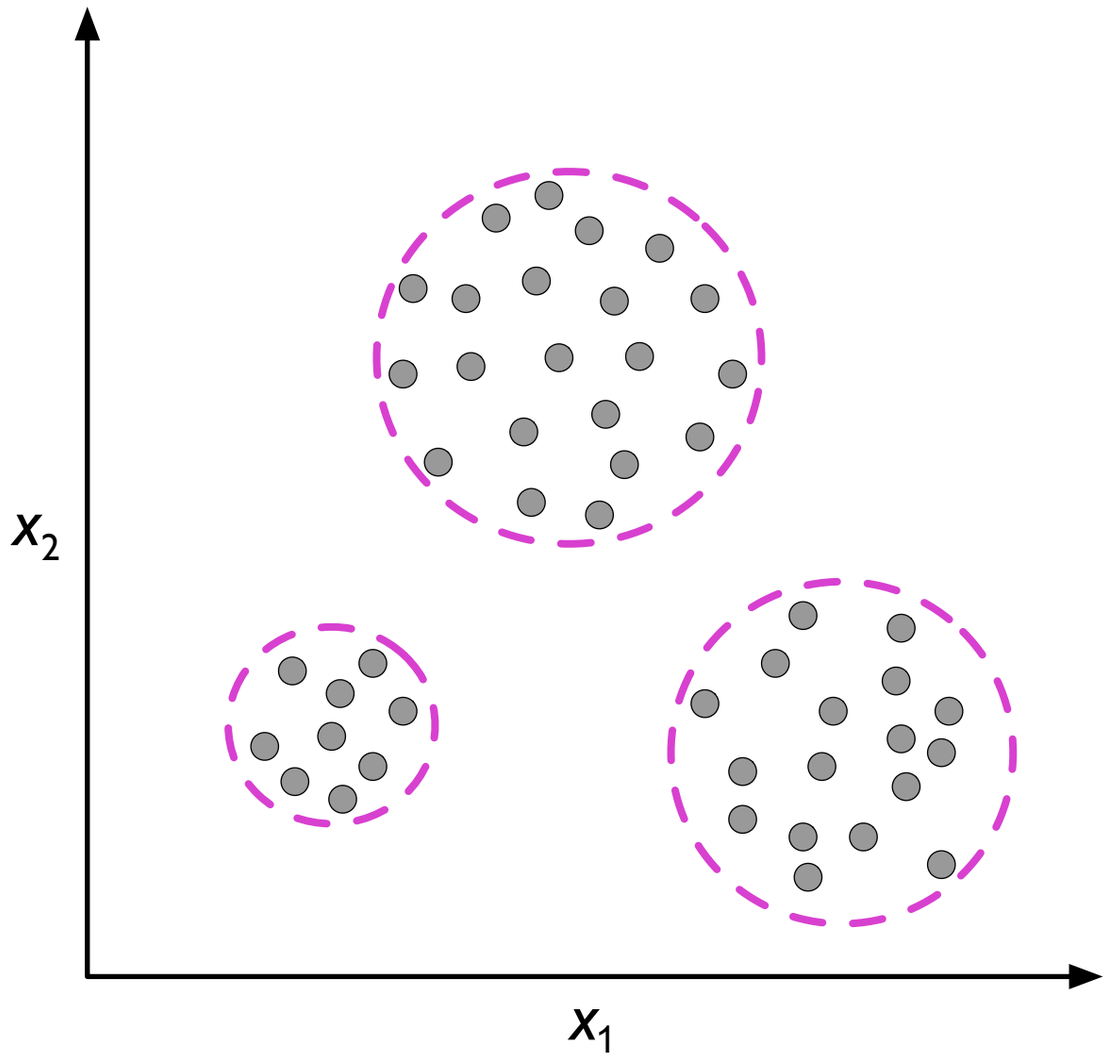
Reminder: Classification works like this



Source: https://3.bp.blogspot.com/-OUd11VBJNAM/VsFacR_YhBI/AAAAAAAABh0/ZKfKAnRj3x0/s1600/cannot%2Bresist.jpg

Unsupervised Learning 2: Clustering

Assigning group memberships to unlabelled examples (instances, data points)



Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

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

www.nature.com/scientificreports/

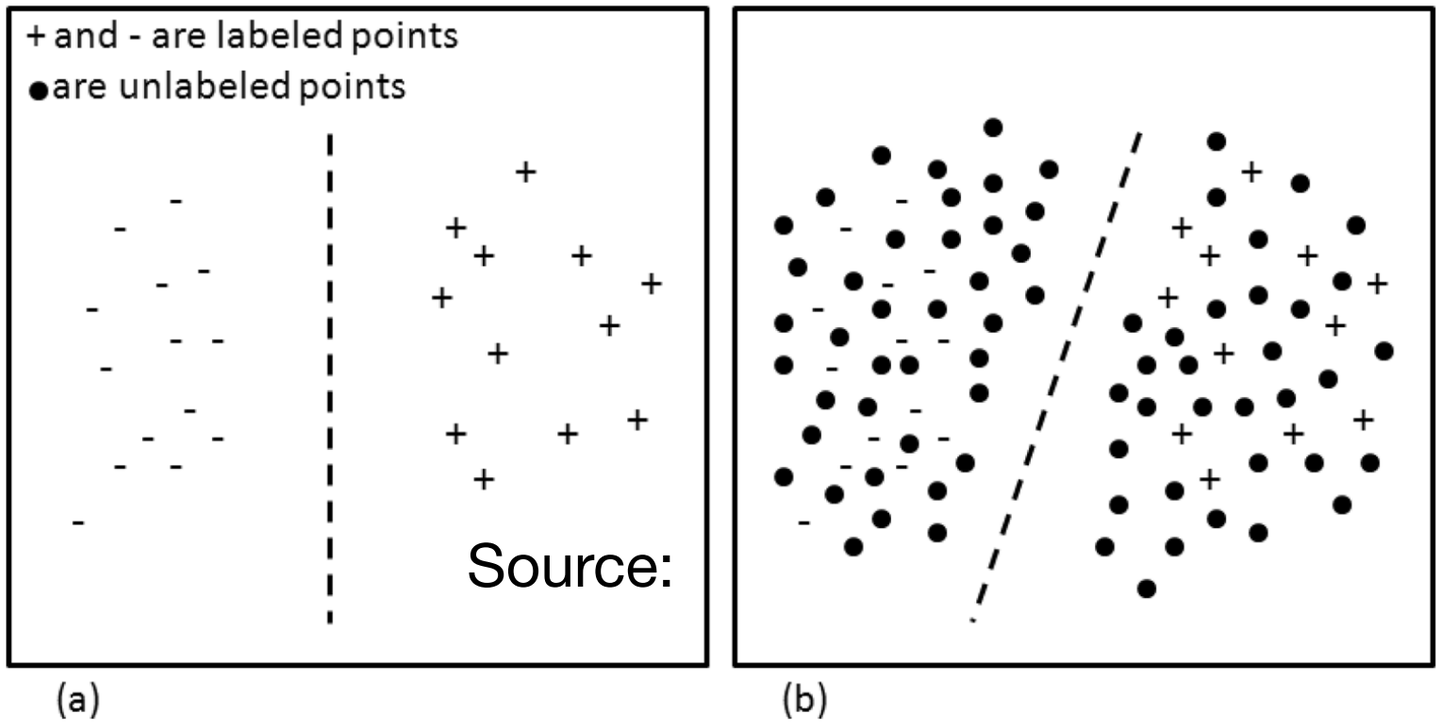


Figure 1. Semi-supervised learning tries to increase the generalization of classification performance by placing the decision boundary through the sparse regions in presence of both labeled and unlabeled data points. (**a**) The decision boundary in presence of labeled data points only, and (**b**) the decision boundary in presence of both labeled and unlabeled data.

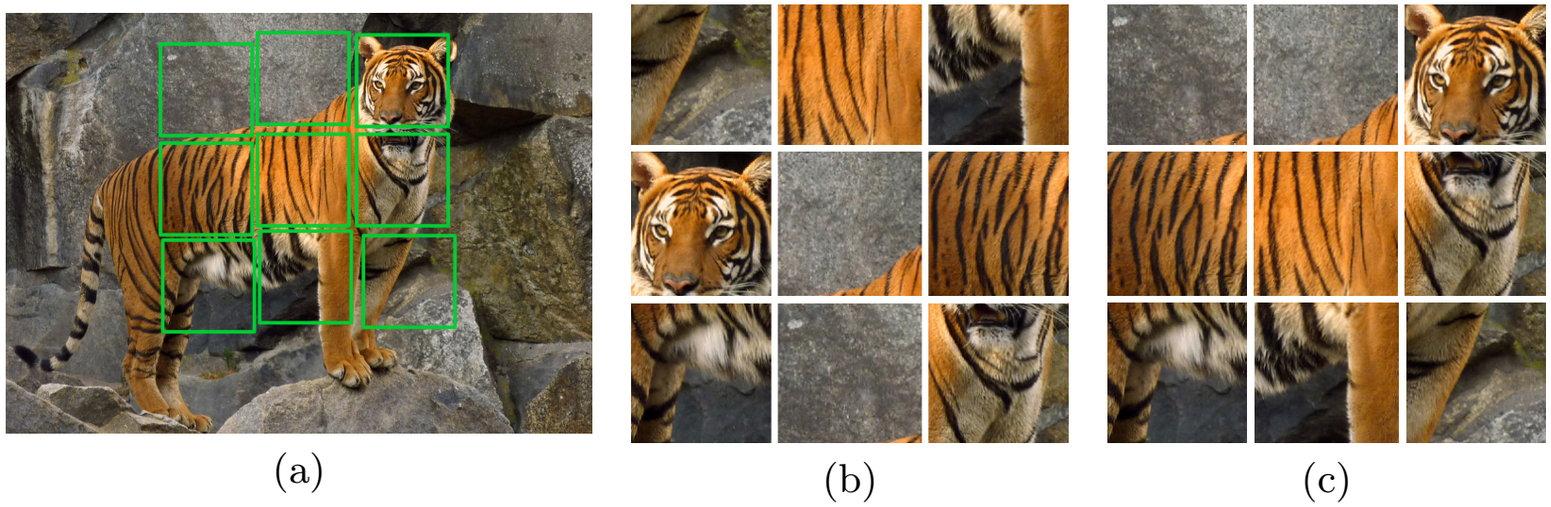
In this paper, we present a semi-supervised learning method that analyzes groups of labeled and unlabeled points in multidimensional feature space in order to identify areas of high density and then guides the learning method to place decision boundaries through the regions with low density. We apply this technique to the analysis of digital pathology images of breast cancer.

Source: Peikari, M., Salama, S., Nofech-Mozes, S., & Martel, A. L. (2018). A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, *8*(1), 1-13.

Self-Supervised Learning

- A recent development and promising research trend in deep learning
- particularly useful if pre-trained models for transfer learning are not available for the target domain
- a process of deriving and utilizing label information directly from the data itself rather than having humans annotating it

Self-Supervised Learning



(a) (b) (c)

Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].

Source: Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." In *European Conference on Computer Vision*, pp. 69-84. Springer, Cham, 2016.

Reinforcement Learning:
The third subcategory of ML (and DL)

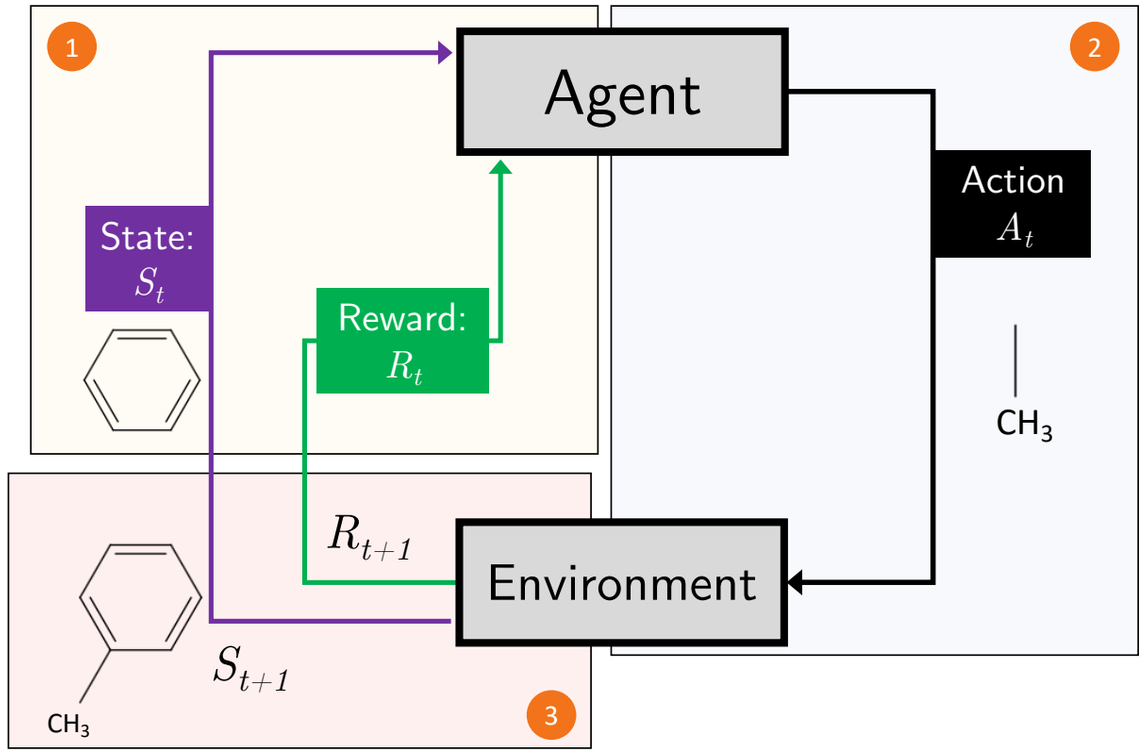


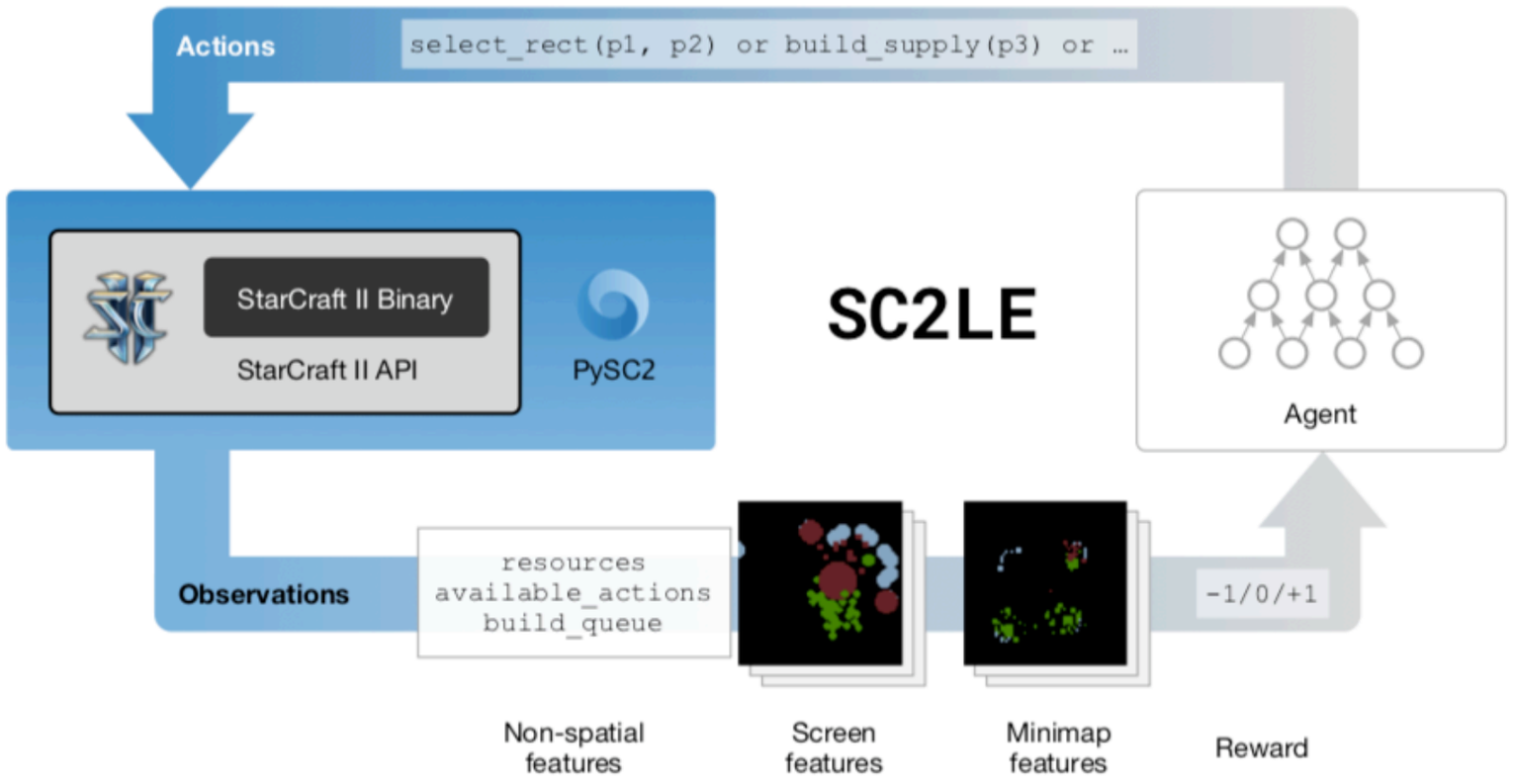
Figure 5: Representation of the basic reinforcement learning paradigm with a simple molecular example. (1) Given a benzene ring (state S_t at iteration t) and some reward value R_t at iteration t, (2) the agent selects an action A_t that adds a methyl group to the benzene ring. (3) The environment considers this information for producing the next state (S_{t+1}) and reward (R_{t+1}). This cycle repeats until the episode is terminated.

Source: Sebastian Raschka and Benjamin Kaufman (2020)
Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition

(Won't cover this in this course)

Reinforcement Learning:
The third subcategory of ML (and DL)

Current state-of-the-art benchmark: StarCraft II



Vinyals, Oriol, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani et al. "Starcraft II: A new challenge for reinforcement learning." *arXiv preprint arXiv:1708.04782* (2017).

Machine Learning Terminology and Notation

(Again, this also applies to DL)

1/5 -- What Is Machine Learning?
2/5 -- The 3 Broad Categories of ML
3/5 -- Machine Learning Terminology and Notation
4/5 -- Machine Learning Modeling Pipeline
5/5 --The Practical Aspects: Our Tools!

Machine Learning Jargon 1/2

- ***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
  (previous to DL, extensive feature engineering required)

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

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

Regression

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

Data Representation

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

Feature vector

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

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

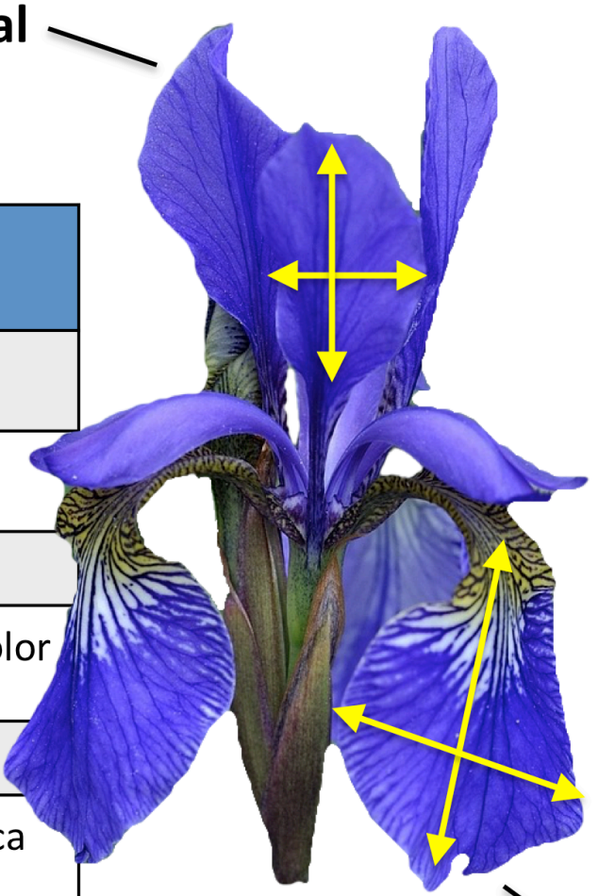
Design Matrix

Data Representation (structured data)

$m =$ ____

$n =$ ____

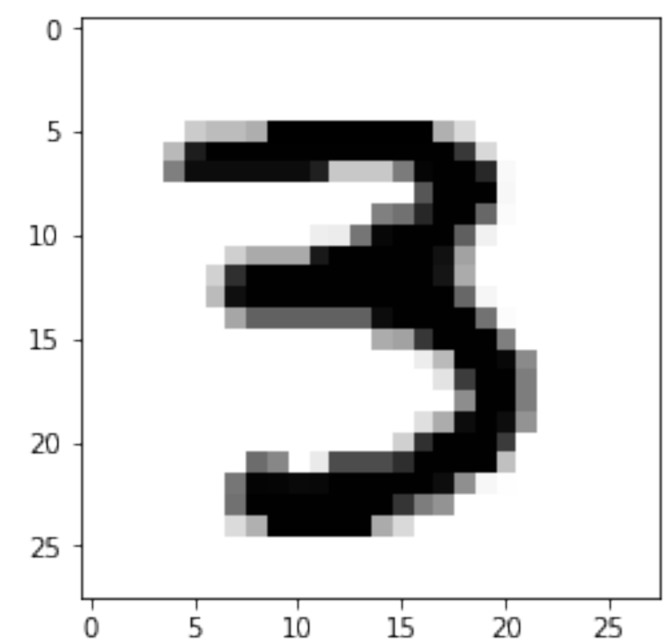
	Sepal length	Sepal width	Petal length	Petal width	
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



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, 0.0, 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, 0.0, 0.5529412, 0.992
15686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.7921569, 0.50980395, 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, 0.14117648, 0.30980393, 0.99215
686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.32941177, 0.14901961, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]



Data Representation (unstructured data; images)

Convolutional Neural Networks

```
Image batch dimensions: torch.Size([128, 1, 28, 28])
Image label dimensions: torch.Size([128])
```

← "NCHW" representation (more on that later)

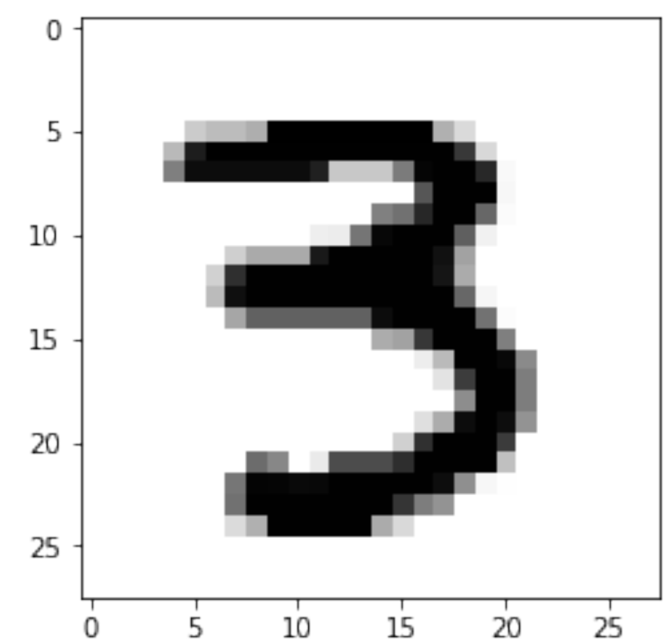
```
print(images[0].size())
```

```
torch.Size([1, 28, 28])
```

```
images[0]
```

```
tensor([[[0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],

         [0.0000, 0.0000, 0.0000, 0.0000, 0.5020, 0.9529, 0.9529, 0.9529,
          0.9529, 0.9529, 0.9529, 0.8706, 0.2157, 0.2157, 0.2157, 0.5176,
          0.9804, 0.9922, 0.9922, 0.8392, 0.0235, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.6627, 0.9922, 0.9922, 0.9922, 0.0314, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.4980, 0.5529,
          0.8471, 0.9922, 0.9922, 0.5961, 0.0157, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0667, 0.0745, 0.5412, 0.9725, 0.9922,
```



Machine Learning Jargon 2/2

- **Training example**, synonymous to
  observation, training record, training instance, training sample (in some contexts, sample refers to a collection of training examples)
- **Feature**, synonymous to
  predictor, variable, independent variable, input, attribute, covariate
- **Target**, synonymous to
  outcome, ground truth, output, response variable, dependent variable, (class) label (in classification)
- **Output / Prediction**, use this to distinguish from targets; here, means output from the model

- use loss L for a single training example
- use cost C for the average loss over the training set
- use $\phi(\cdot)$, unless noted otherwise, for the activation function

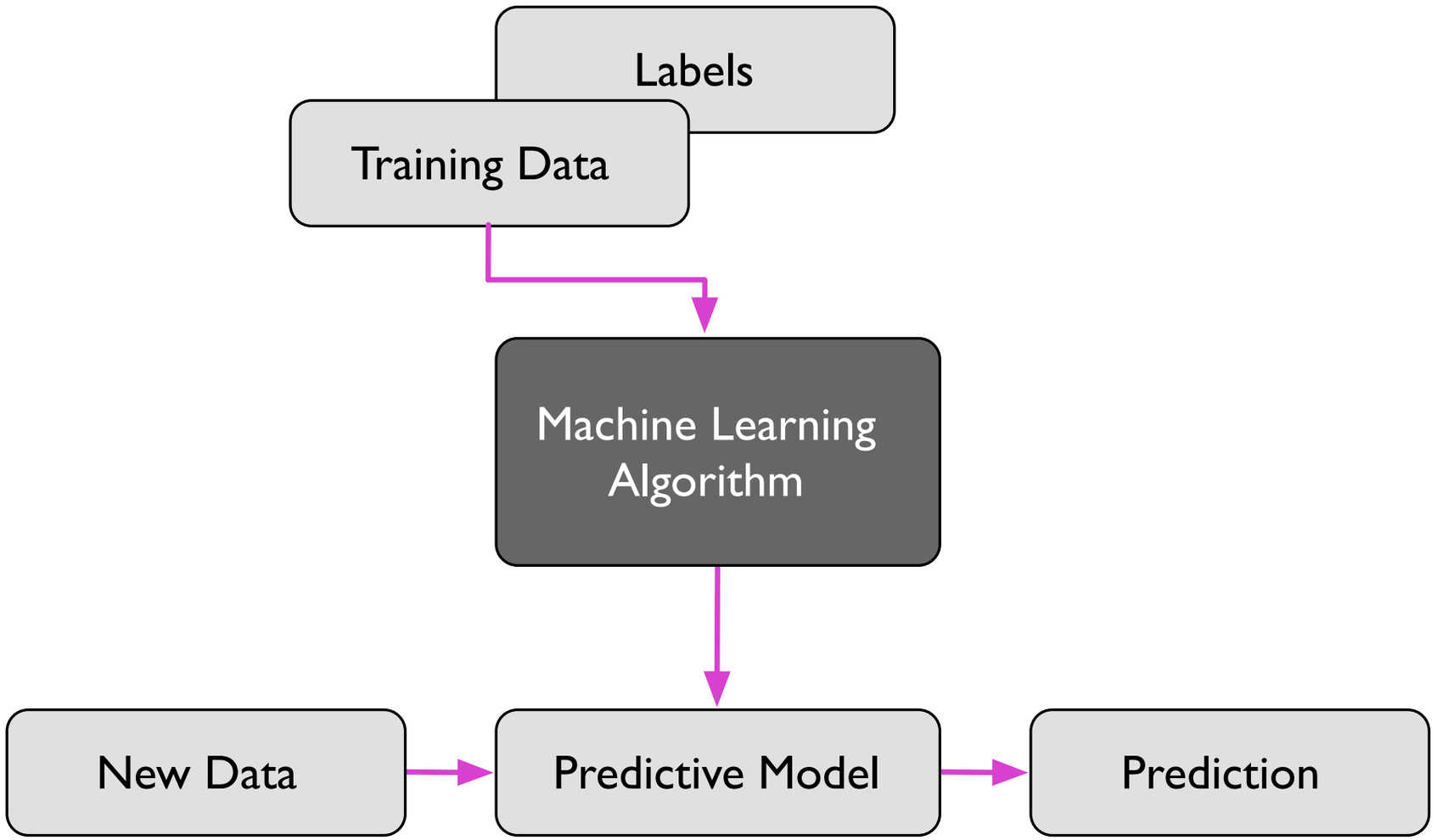
(will make more sense later)

Machine Learning Modeling Pipeline

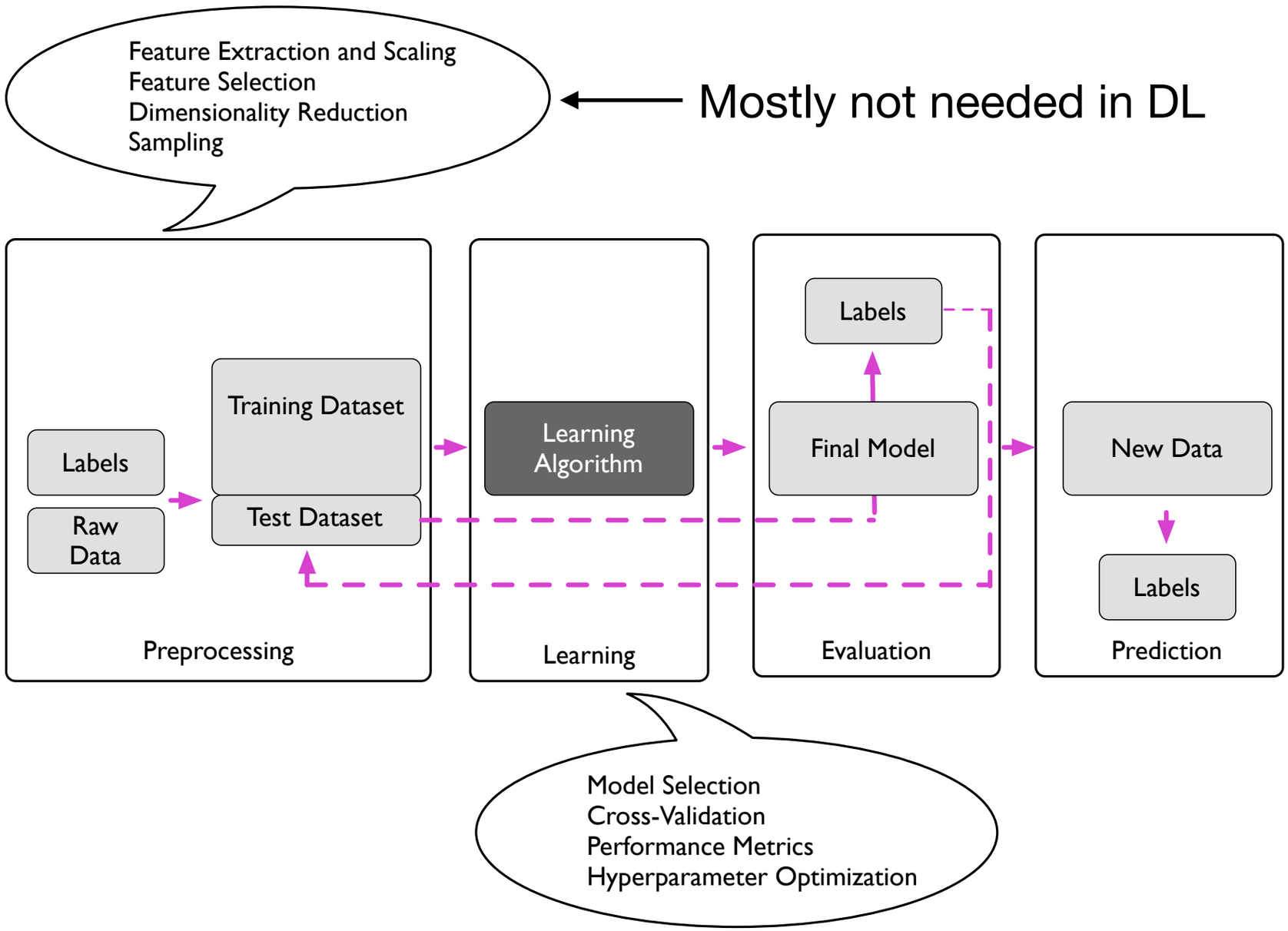
(Like before, this also applies to DL)

1/5 -- What Is Machine Learning?
2/5 -- The 3 Broad Categories of ML
3/5 -- Machine Learning Terminology and Notation
4/5 -- Machine Learning Modeling Pipeline
5/5 --The Practical Aspects: Our Tools!

Supervised Learning Workflow

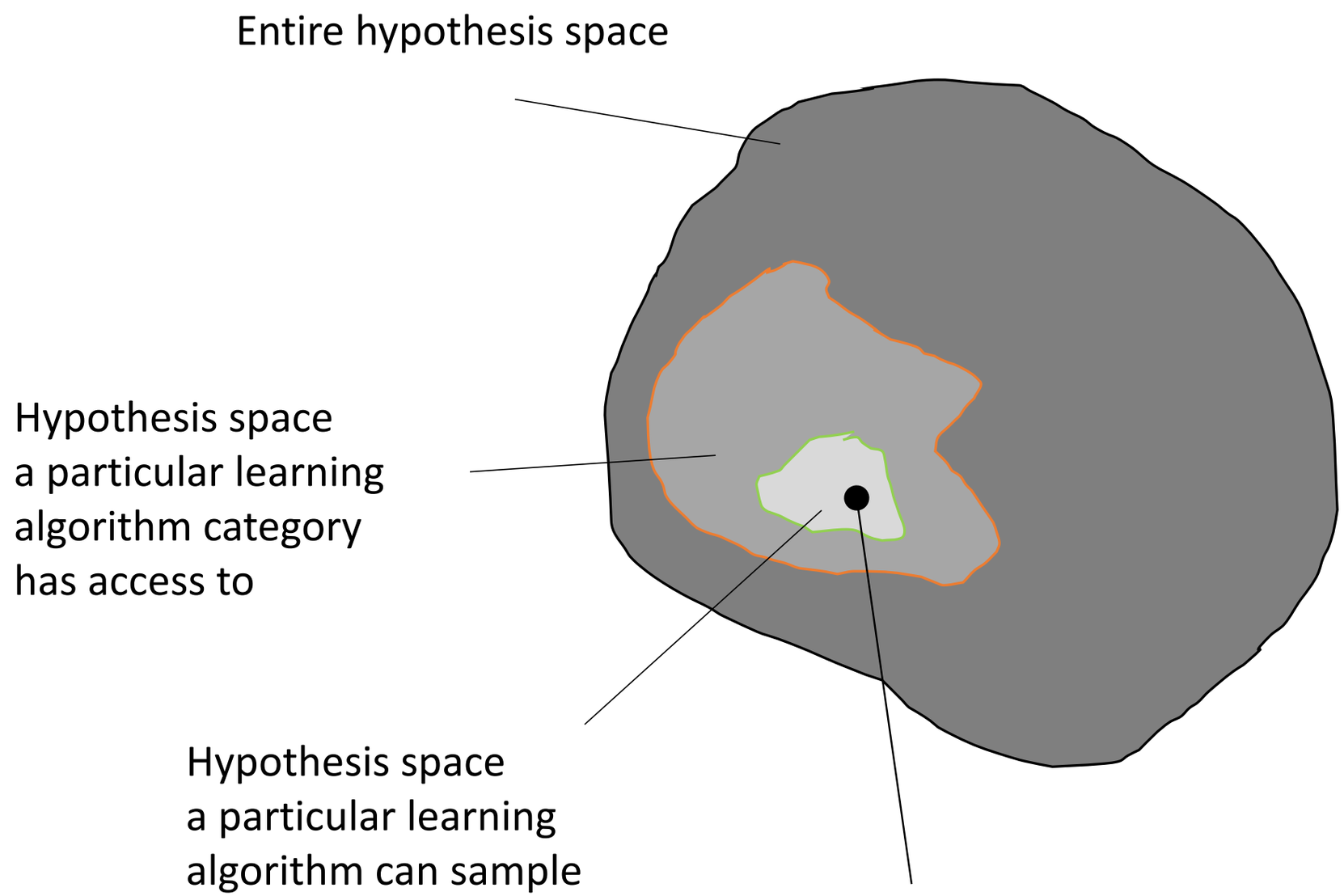


Supervised Learning Workflow (more detailed)



Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

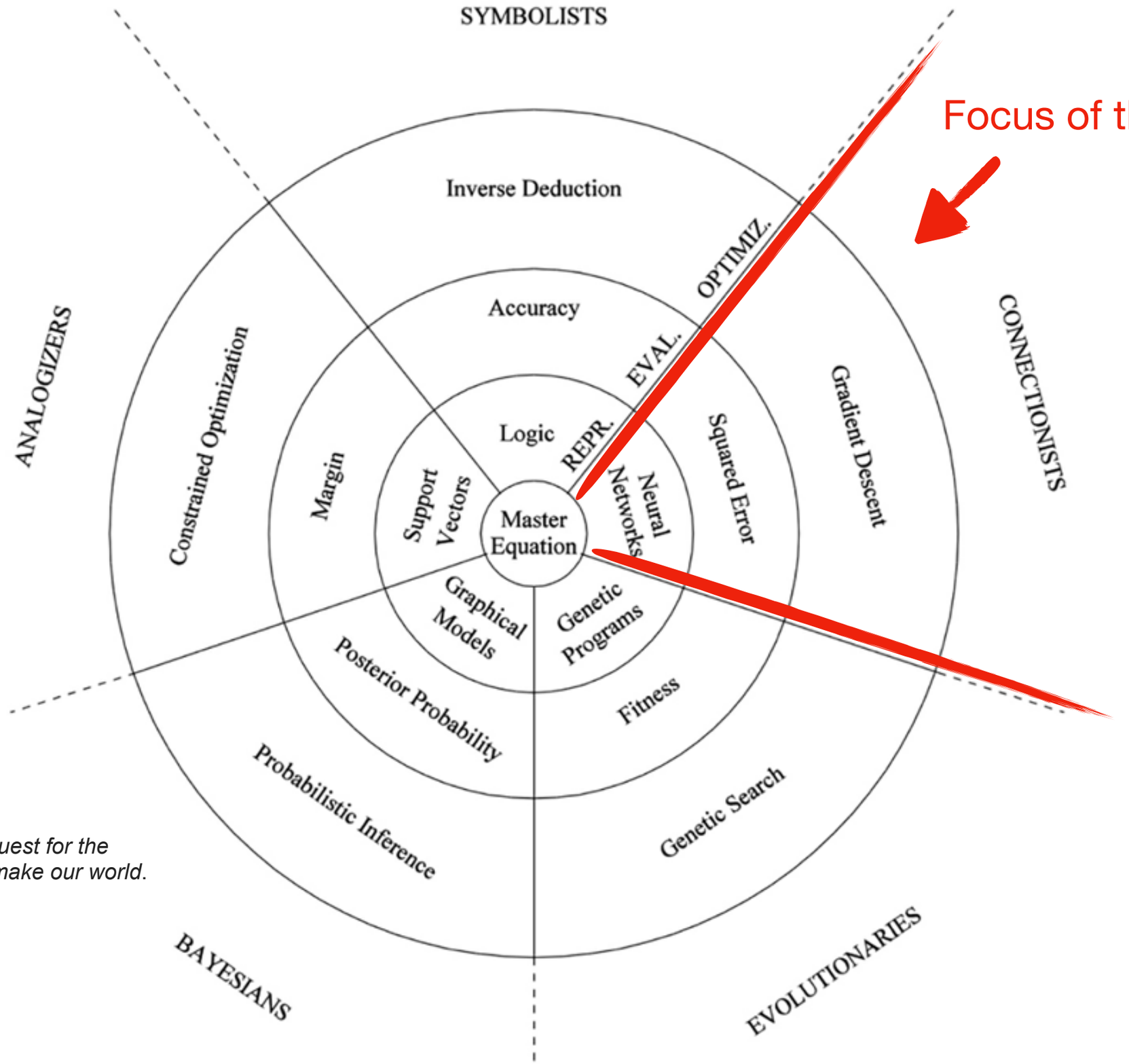
Hypothesis Space



5 Steps for Approaching an ML/DL Problem

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

Pedro Domingo's 5 Tribes of Machine Learning



Source: Domingos, Pedro.
The Master Algorithm: How the quest for the ultimate learning machine will remake our world.
Basic Books, 2015.

Learning = Representation + Evaluation + Optimization

(Pedro Domingos, *A Few Useful Things to Know about Machine Learning* https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf)

Objective Functions / Surrogate Risk/Loss

- Maximize the posterior probabilities (e.g., naive Bayes)
- Maximize a fitness function (genetic programming)
- Maximize the total reward/value function (reinforcement learning)
- Maximize information gain/minimize child node impurities (CART decision tree classification)
- Minimize a mean squared error cost (or loss) function (CART, decision tree regression, linear regression, adaptive linear neurons, ...)
- Maximize log-likelihood or minimize cross-entropy loss (or cost) function
- Minimize hinge loss (support vector machine)

Optimization Methods

- Combinatorial search, greedy search (e.g., decision trees)
- Unconstrained convex optimization (e.g., logistic regression)
- Constrained convex optimization (e.g., SVM)
- Nonconvex optimization, here: using backpropagation, chain rule, reverse autodiff. (e.g., neural networks)
- Constrained nonconvex optimization (e.g., semi-adversarial nets)

0/1 Loss, Misclassification Error

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

$$ERR_{\mathscr{D}}\textbf{test} = \frac{1}{n}\sum_{i=1}^{n} L\left(\hat{y}^{[i]}, y^{[i]}\right)$$

Other Performance Metrics

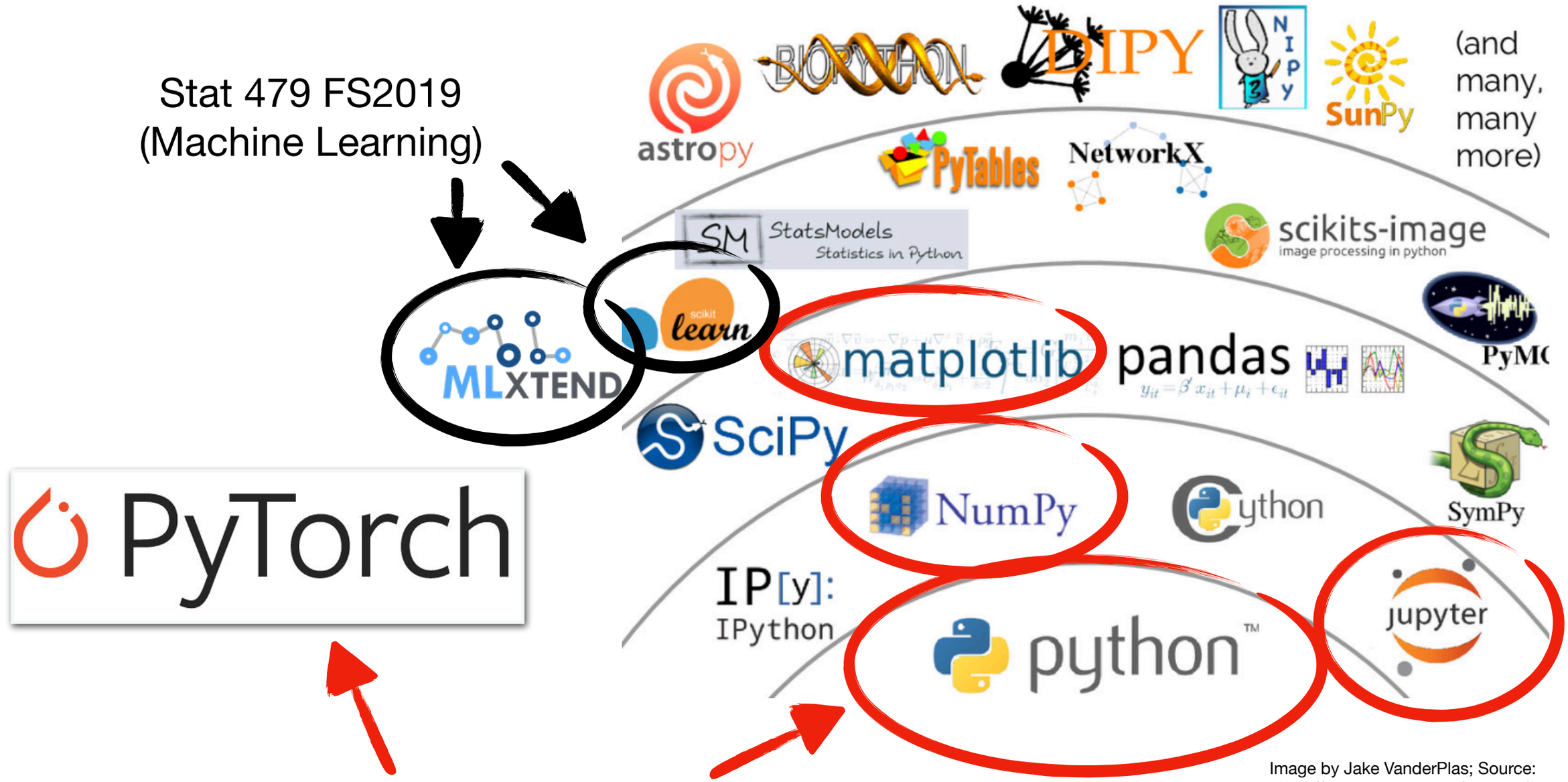
- Accuracy (1-Error)
- ROC AUC
- Precision
- Recall
- (Cross) Entropy
- Likelihood
- Squared Error/MSE
- L-norms
- Utility
- Fitness
- ...

The Practical Aspects: Our Tools!

1/5 -- What Is Machine Learning?
2/5 -- The 3 Broad Categories of ML
3/5 -- Machine Learning Terminology and Notation
4/5 -- Machine Learning Modeling Pipeline
5/5 --The Practical Aspects: Our Tools!

Main Scientific Python Libraries

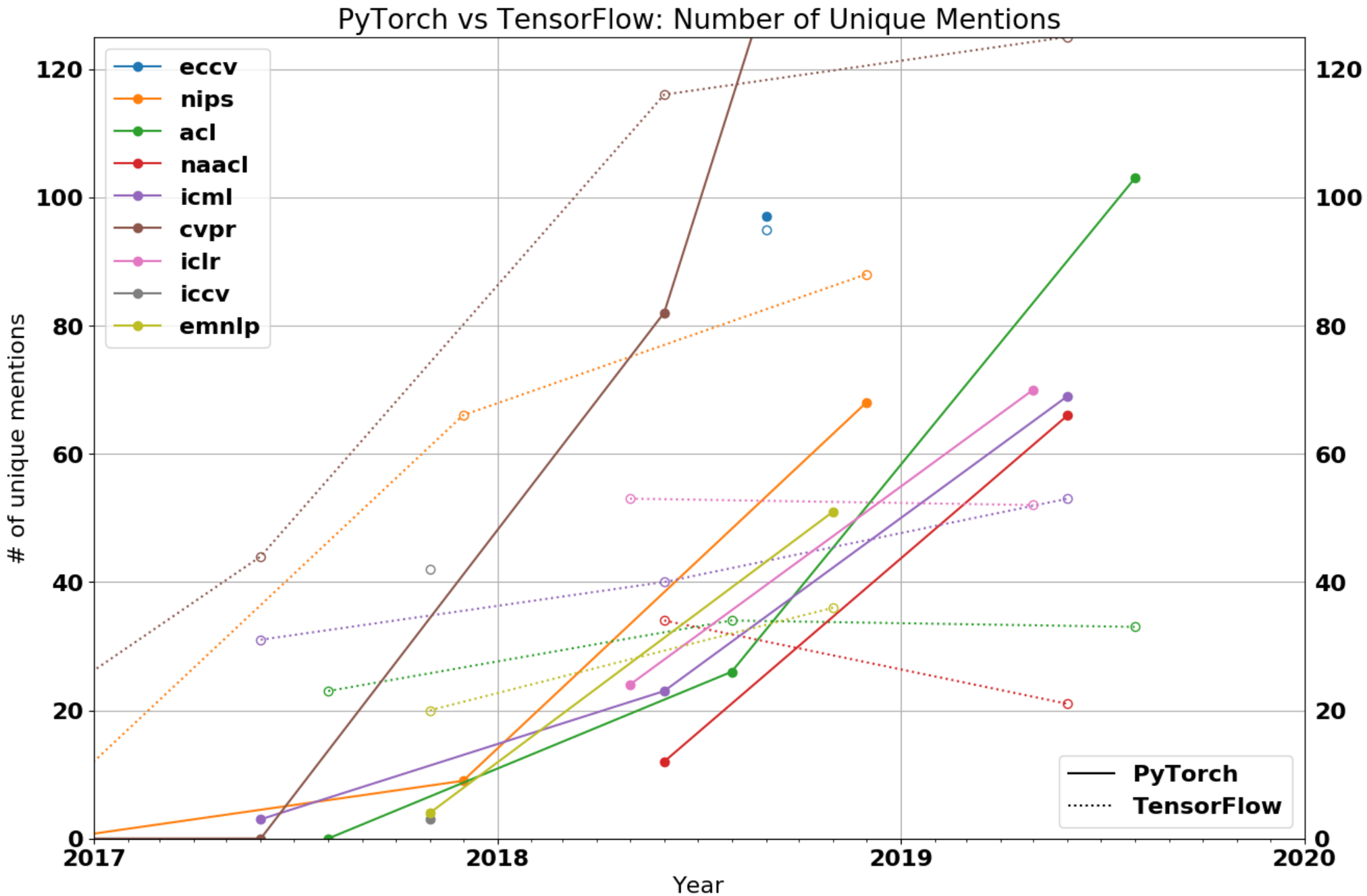
Stat 479 FS2019
(Machine Learning)



Main tools for this course

Image by Jake VanderPlas; Source: https://speakerdeck.com/jakevdp/the-state-of-the-stack-scipy-2015-keynote?slide=8)

"The State of Machine Learning Frameworks in 2019"



Source:
https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/

"The State of Machine Learning Frameworks in 2019"

CONFERENCE	PT 2018	PT 2019	PT GROWTH	TF 2018	TF 2019	TF GROWTH
CVPR	82	280	240%	116	125	7.7%
NAACL	12	66	450%	34	21	-38.2%
ACL	26	103	296%	34	33	-2.9%
ICLR	24	70	192%	54	53	-1.9%
ICML	23	69	200%	40	53	32.5%

In 2018, PyTorch was a minority. Now, it is an overwhelming majority, with 69% of CVPR using PyTorch, 75+% of both NAACL and ACL, and 50+% of ICLR and ICML. While PyTorch's dominance is strongest at vision and language conferences (outnumbering TensorFlow by 2:1 and 3:1 respectively), PyTorch is also more popular than TensorFlow at general machine learning conferences like ICLR and ICML.

Source:
https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/

Next Lecture:

A Brief Summary of the History of Neural Networks and Deep Learning

Reading Assignments

- Pedro Domingos, *A Few Useful Things to Know about Machine Learning* https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

- STAT479 FS2019: Machine Learning, lecture notes 01: https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/01_overview/01-ml-overview__notes.pdf

(exam questions also assume that you read the assigned reading materials)