Learning from Data

Hsi-Pin Ma 馬席彬

http://lms.nthu.edu.tw/course/40724

Department of Electrical Engineering

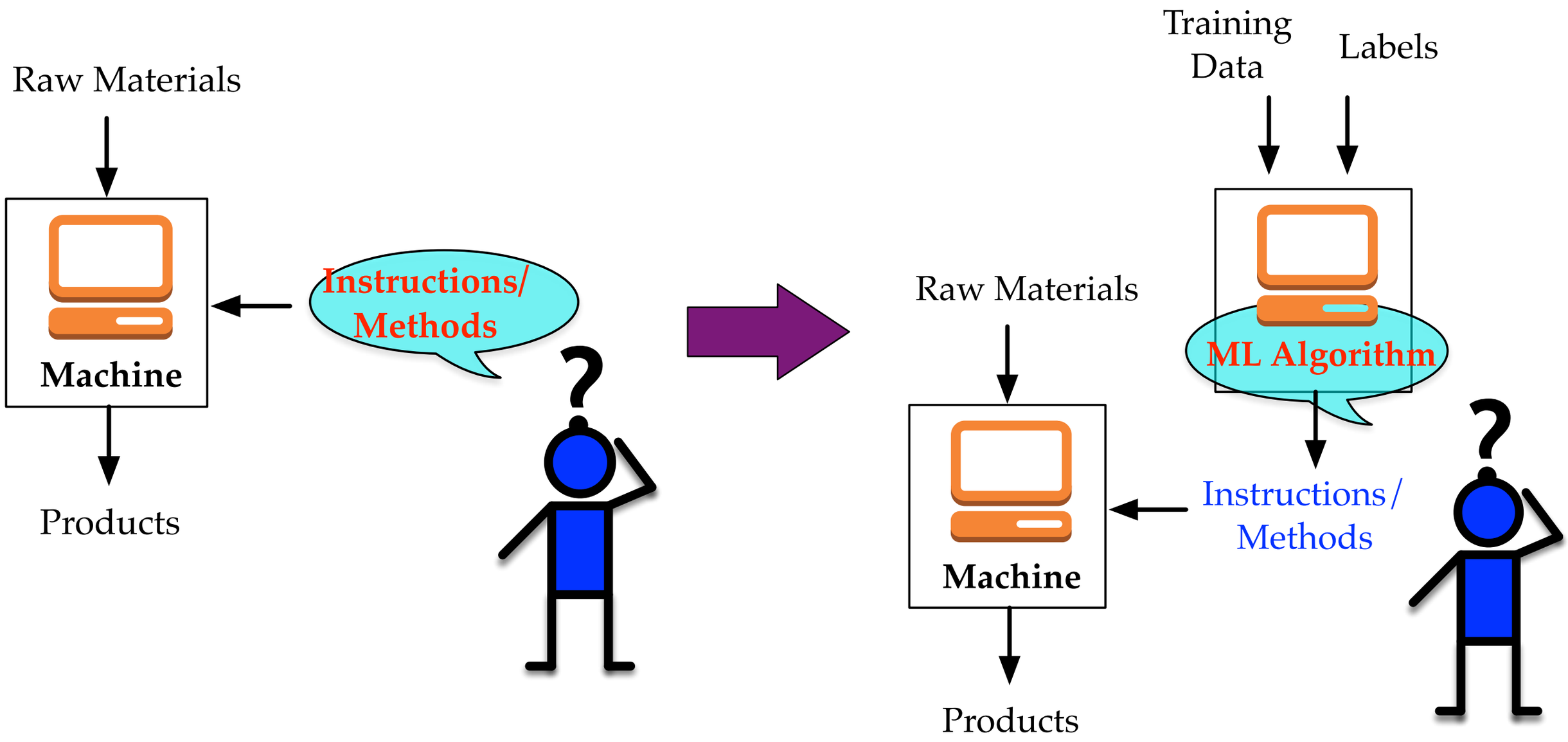
National Tsing Hua University

Outline

- General Concepts of Machine Learning
- Three Types of Machine Learning
- Building Blocks for Machine Learning Systems
- Using Python for Machine Learning

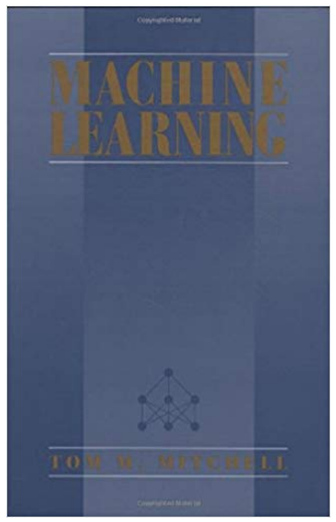
General Concepts of Machine Learning

Progress to Use Machine to Assist or Replace Human's Works



Machine Learning

- The field of machine learning is concerned with the question of *how to construct computer programs* that *automatically improve with experience*.



- 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*.

by Tom M. Mitchell

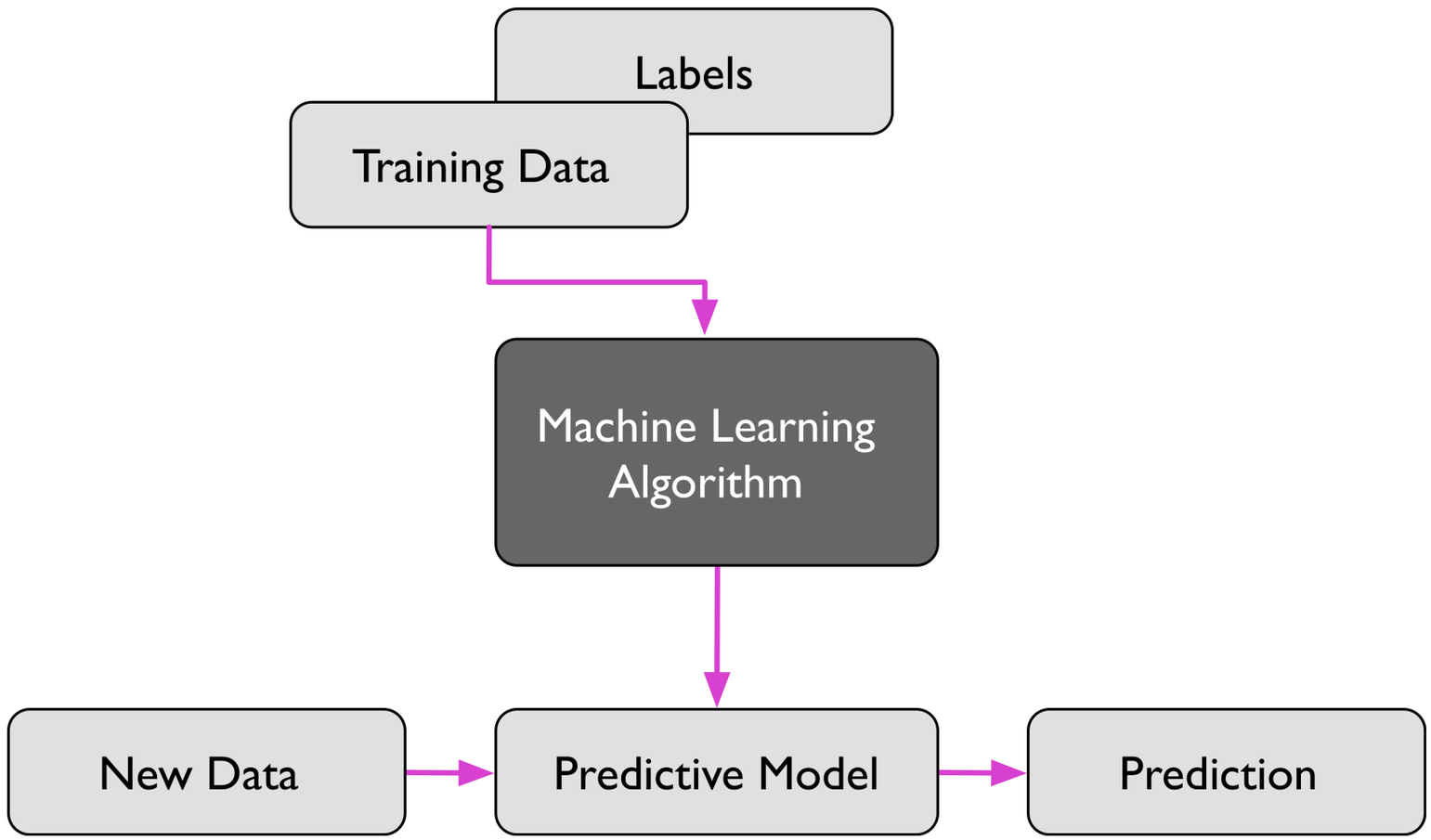
Three Types of Machine Learning

Three Types of Machine Learning

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

Supervised Learning

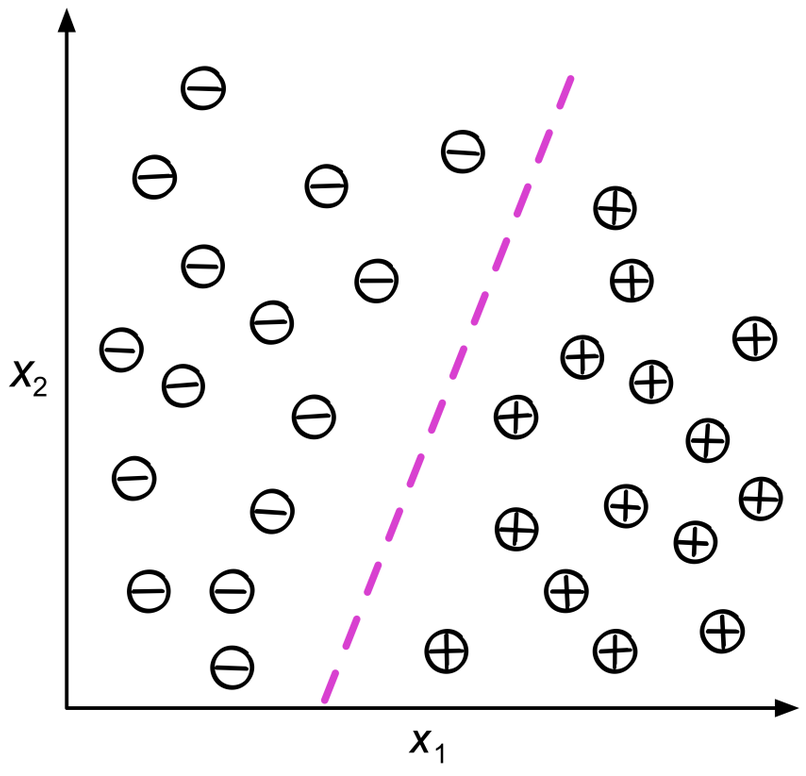
- Making predictions about the future
  - To learn a model from labeled training data that allows us to make predictions about unseen or future data



  - Classification vs. Regression

Classification

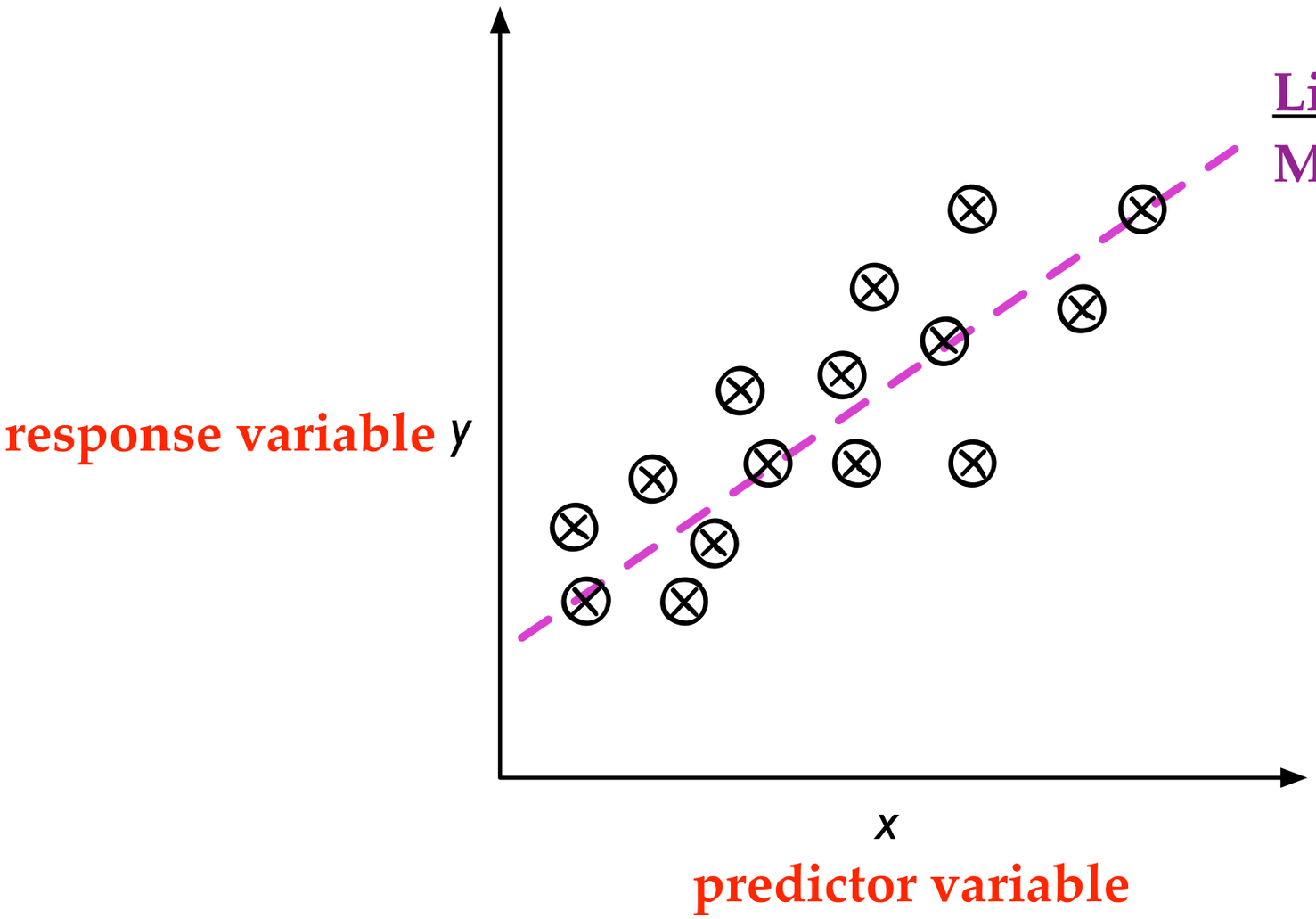
- Predict the class labels of new instance based on past observations. The class labels are ***discrete unordered*** values.
  - Binary: spam mail detection
  - Multiclass: handwritten character recognition



x_1, x_2: features of data items; +.-: labels

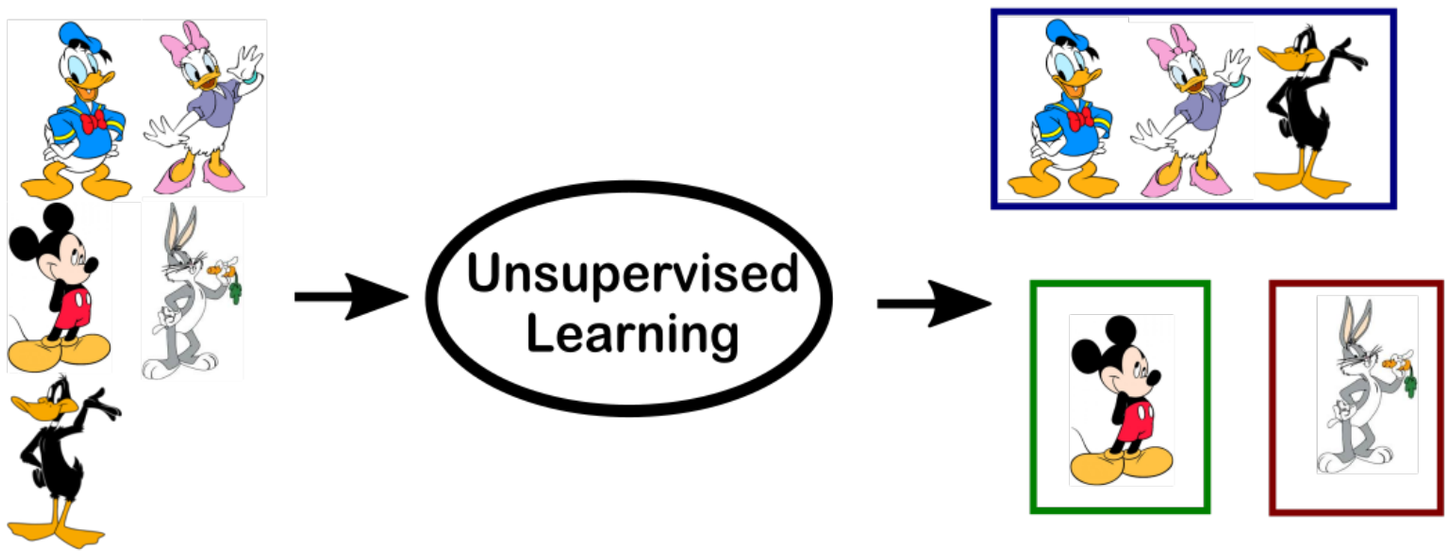
Regression

- Prediction of ***continuous*** outcomes
  - Fit a function to a set of real data points



Unsupervised Learning

- Deal with ***unlabeled*** data or data of ***unknown*** structure
  - Use unsupervised learning to ***explore the structure*** of the data to extract meaningful information without guidance

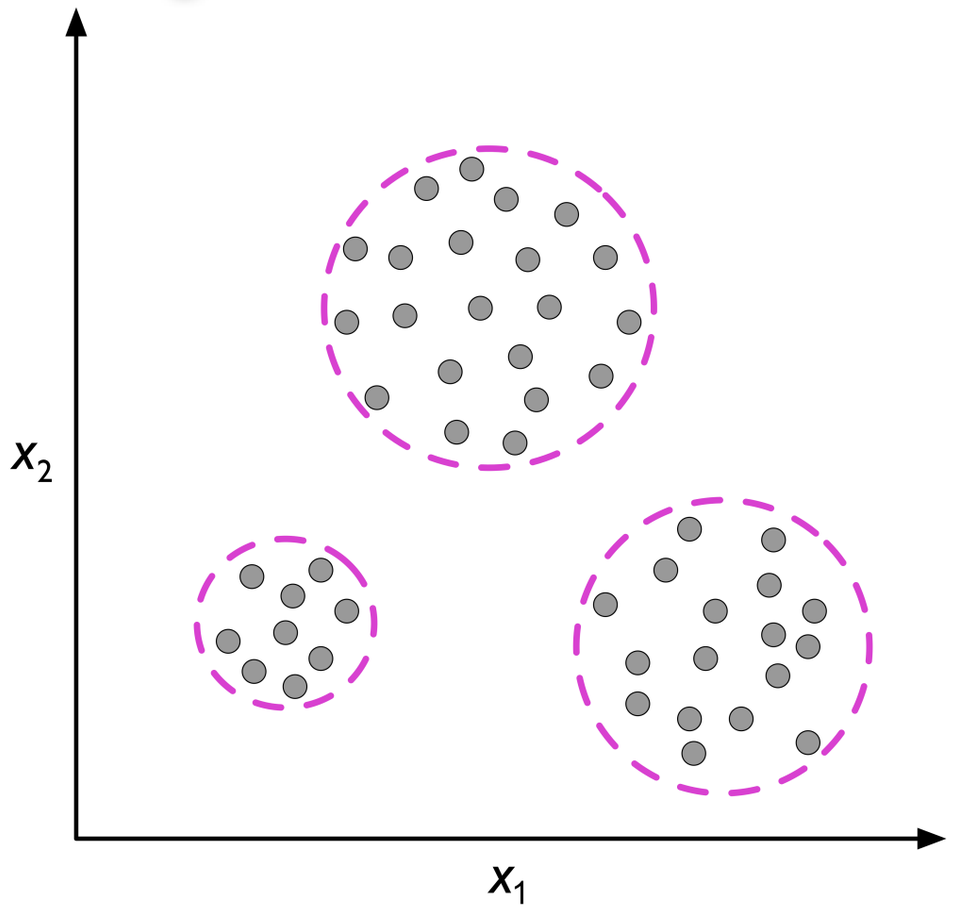


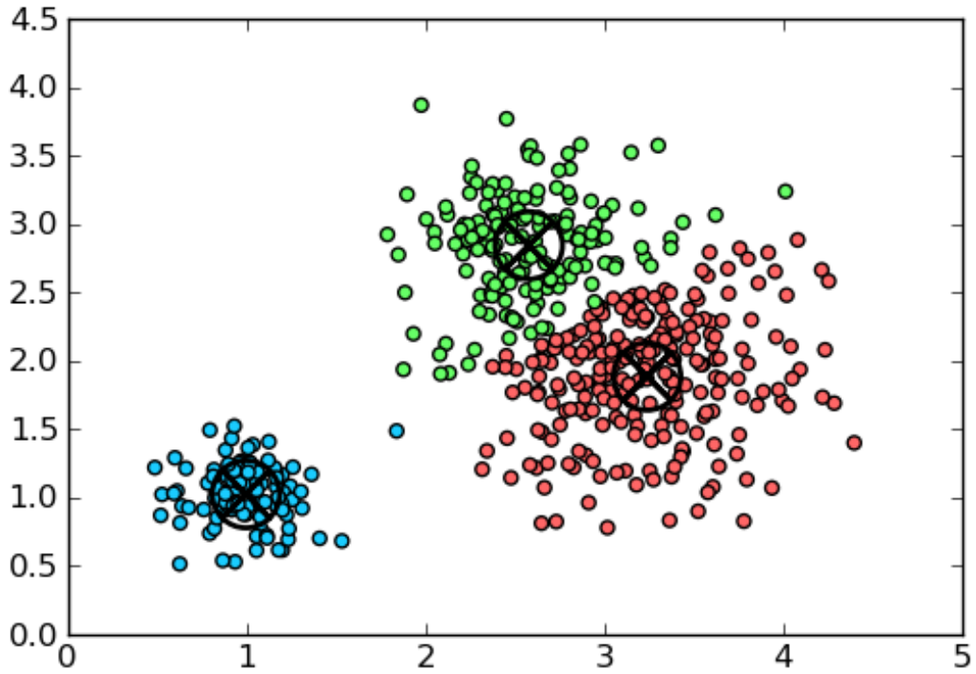
[Devin Soni]

- Clustering vs. dimension reduction

Clustering

- An exploratory data analysis technique that allows to organize a pile of information into meaningful subgroups (clusters) without any prior knowledge
- Objects within a cluster share a degree of similarity (unsupervised classification)

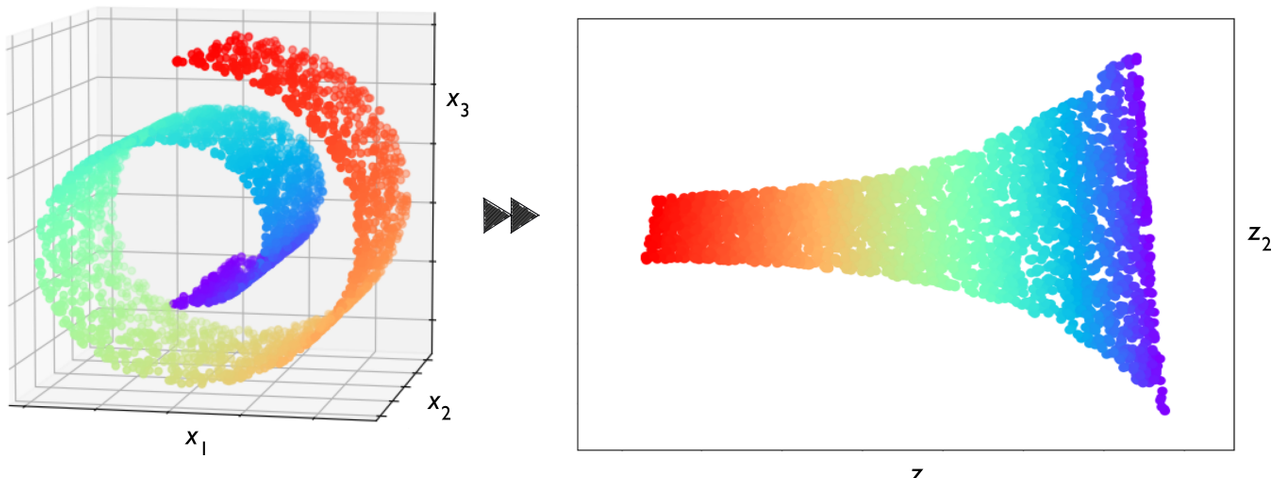


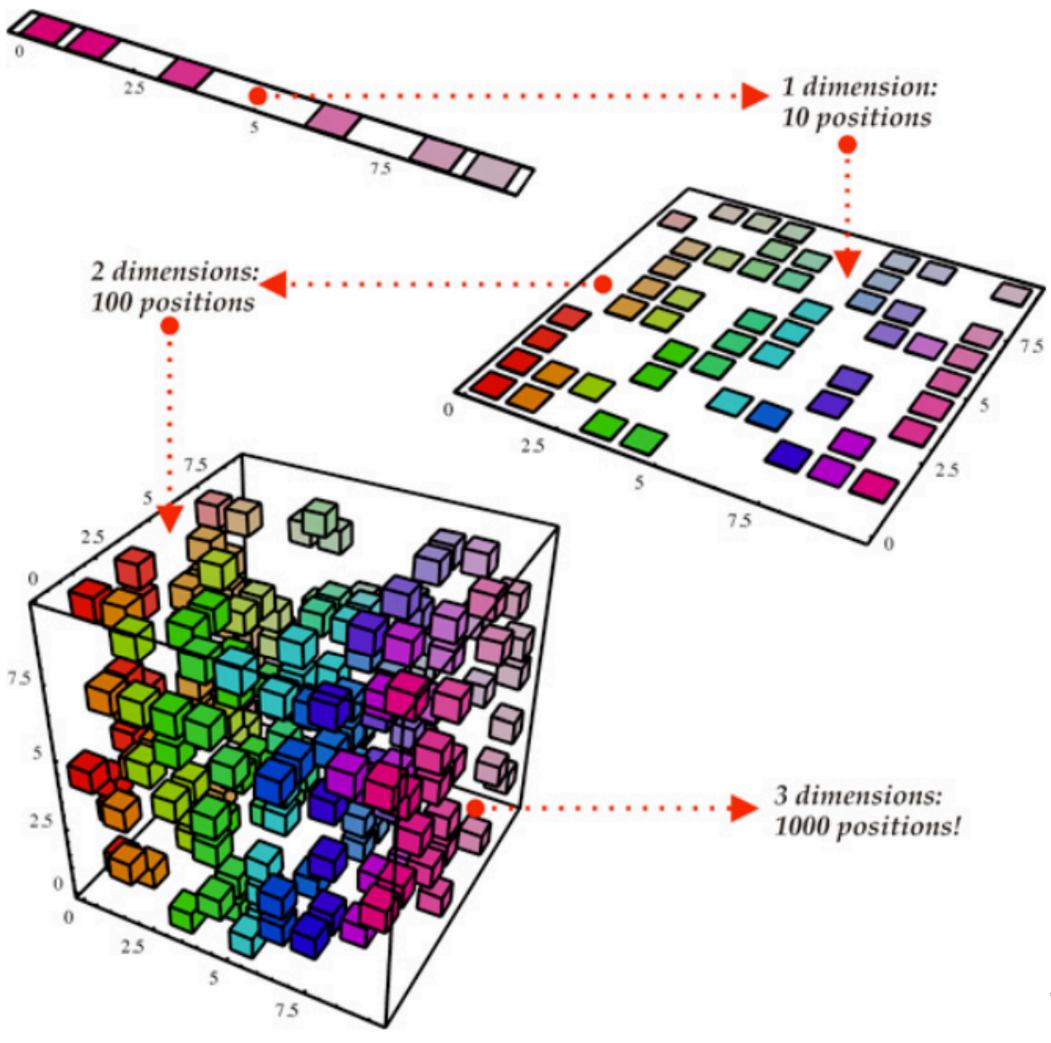


Grouping based on the two features x_1, x_2 of data points

Dimension Reduction

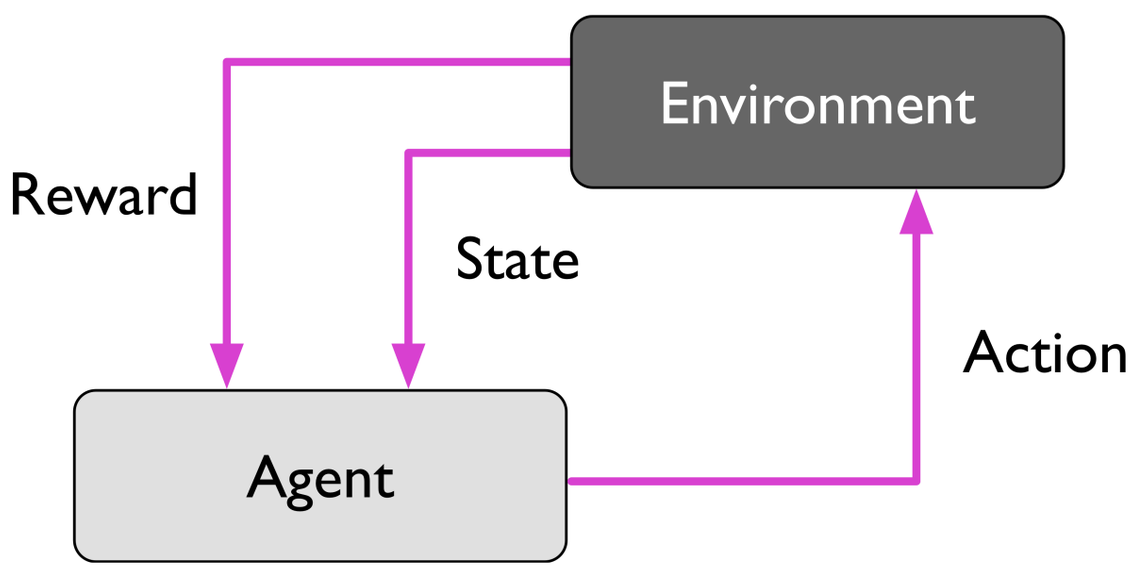
- Detect and identify lower-dimensional structure in higher-dimensional data
  - Represent data using less columns or features without losing too much information
  - (Learning algorithms) running much faster and taking up less memory space
- Data visualization





Reinforcement Learning

- To develop a system (agent) that improves its performance based on interactions with environment
  - The ***agent*** observes the ***state*** of the environment, select and perform ***actions***, and get the ***rewards*** in return
  - The agent must learn by itself what the best ***policy*** is to get the most reward over time.
  - A policy defines what action the agent should take when it is in a given situation

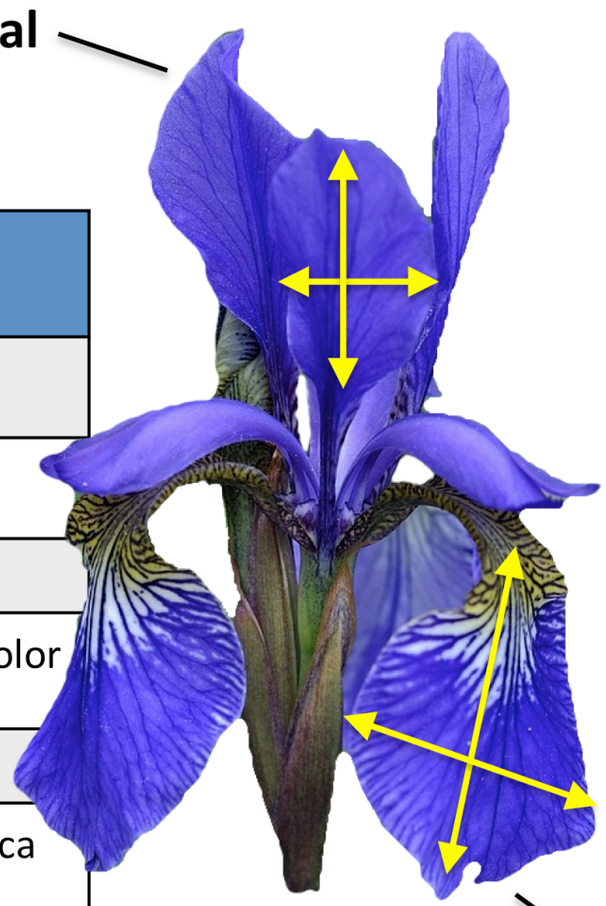


Basic Terminology (1/2)

- Iris dataset

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)

[Fisher, 1936]

[UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/iris]

Basic Terminology (2/2)

- Measurements of 150 iris flowers (150 samples/4 features)
- From 3 different species (Setosa, Versicolor, Virginica)
- Row are samples, and columns are features
- 150 x 4 feature matrix $\mathbf{X} \in \mathbb{R}^{150 \times 4}$
- Each row (one flower instance) $x^{(i)} \in \mathbb{R}^{1 \times 4}$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

$$x^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

- Each feature dimension $x_j \in \mathbb{R}^{150 \times 1}$

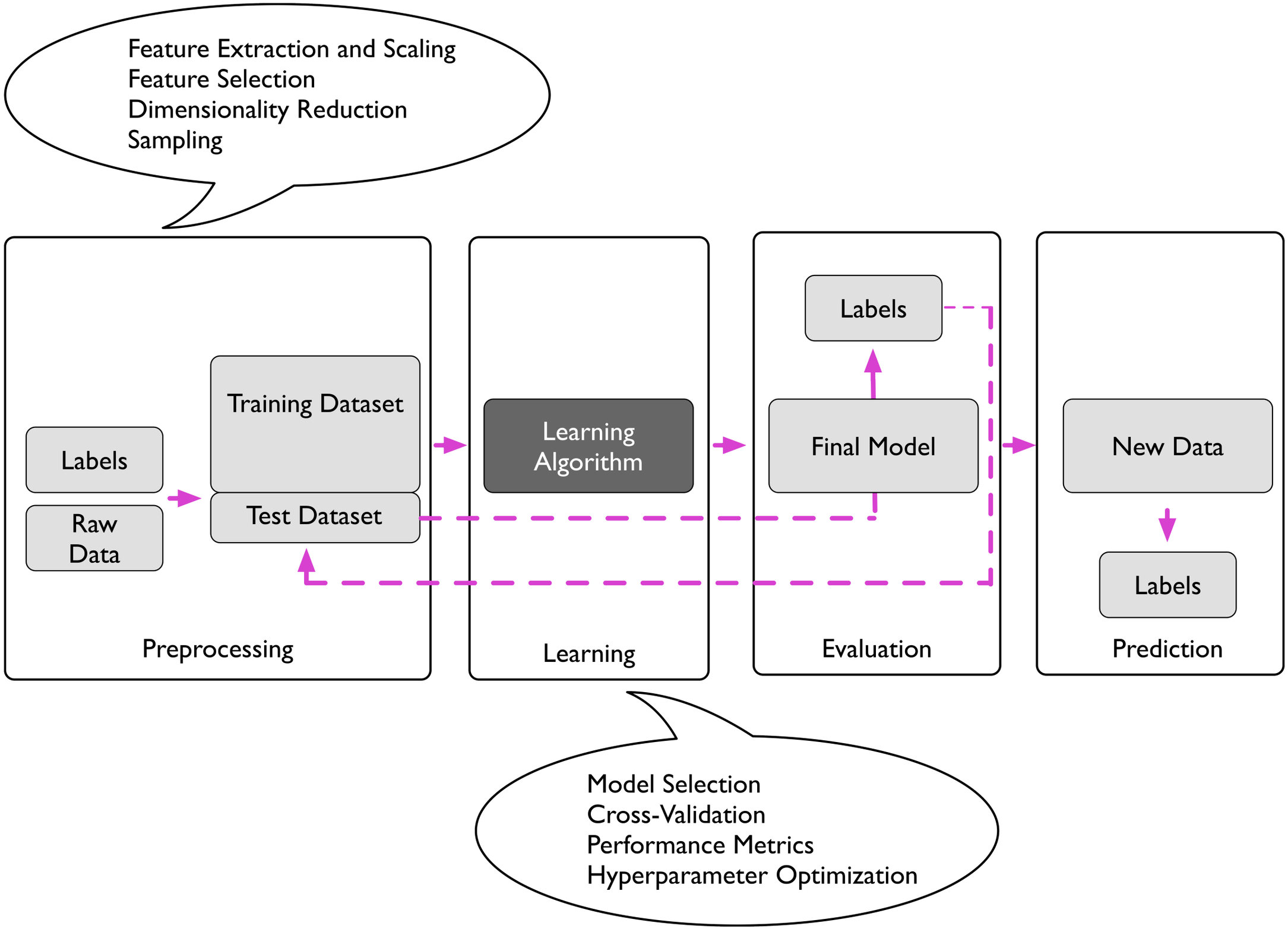
$$x_j = \begin{bmatrix} x_j^1 \\ x_j^2 \\ \vdots \\ x_j^{150} \end{bmatrix}$$

- Target variables (class labels)

$$y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{150} \end{bmatrix} \quad (y \in \{\text{Setosa, Versicolor, Virginica}\})$$

Building Blocks for Machine Learning Systems

A Roadmap for Building ML Systems



Preprocessing - Getting Data into Shape

- Feature extraction and scaling
- Feature selection and dimension reduction
- Dividing data into a training set (train and optimize the model) and a testing set (evaluate the model performance) by random sampling

Learning - Training and Selecting a Predictive Model

- No free lunch theorems
  - We cannot get learning "for free".
  - No *a prior* model guaranteed to work better. The only way is to evaluate all models to find the best. (But not possible)
  - In practice, we make some reasonable assumptions about the data and evaluate only a few models. (Empirically)
- How do we know what model works best?
  - Classification accuracy
  - Cross-validation
  - Hyperparameter optimization to fine tune the model

Evaluating Models and Predicting Unseen Data Instances

- Performance measures
  - Accuracy (errors)
  - Mean square error
- Operations (feature scaling or dimension reduction) applied to training dataset should also be applied to test dataset and new data samples
  - The performance measured on the test data may be overly optimistic otherwise

Using Python for Machine Learning

Python

- A high-level, general purpose, interpreted programming language that is widely used in scientific computing and engineering
- Ecosystem
  - Environments
    - IDLE, **Jupyter** notebook, Spider
  - Version
    - Python 2, **Python 3 (used in the class)**
  - Packages
    - SciPy, NumPy, Matplotlib, pandas, SciKit-Learn, TensorFlow
  - Systems and system libraries
    - os, BLAS, LAPACK

Top Ten Programming Languages

2017

Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise	100.0
2. C	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	99.5
4. C++	Mobile, Enterprise, Embedded	97.1
5. C#	Web, Mobile, Enterprise	87.7
6. R	Enterprise	87.7
7. JavaScript	Web, Mobile	85.6
8. PHP	Web	81.2
9. Go	Web, Enterprise	75.1
10. Swift	Mobile, Enterprise	73.7

2018

Language Types (click to hide)

Web | Mobile | Enterprise | Embedded

Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise, Embedded	100.0
2. C++	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	97.5
4. C	Mobile, Enterprise, Embedded	96.7
5. C#	Web, Mobile, Enterprise	89.4
6. PHP	Web	84.9
7. R	Enterprise	82.9
8. JavaScript	Web, Mobile	82.6
9. Go	Web, Enterprise	76.4
10. Assembly	Embedded	74.1

IEEE Spectrum: https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018