HPC-AI
Introduction to Machine Learning

2019 – 2020

Chloé-Agathe Azencott
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Inserm
Institut national de la santé et de la recherche médicale

institutCurie

PSL RESEARCH UNIVERSITY PARIS

MINES ParisTech

- **Course material & contact**

  `http://cazencott.info`

  `chloe-agathe.azencott@mines-paristech.fr`

  Slides thanks to Ethem Alpaydi, Matthew Blaschko, Trevor Hastie, Rob Tibshirani and Jean-Philippe Vert.

What is (Machine) Learning

Why Learn?

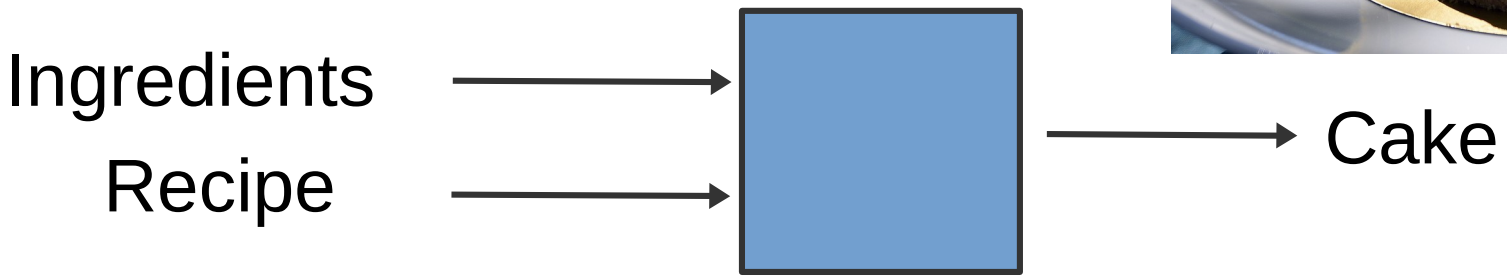
- **Learning**:

  Acquire a **skill** based on **experience**

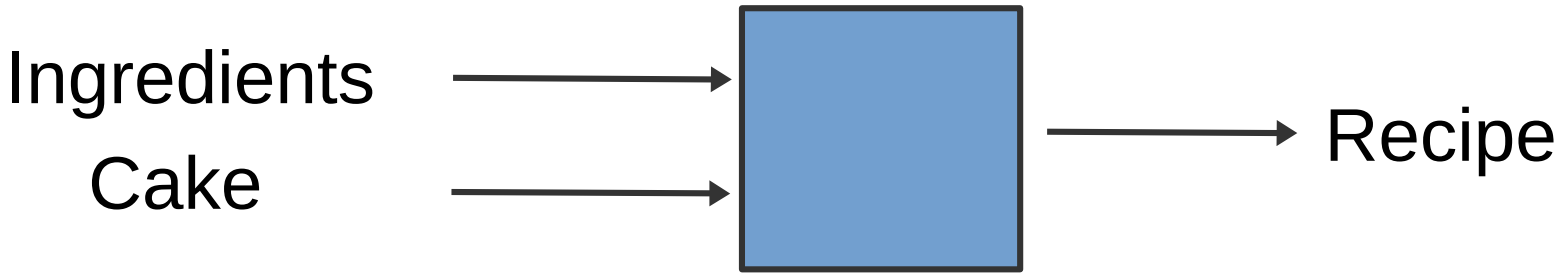
- **Machine learning**:
  - develop an **algorithm/model**
  - using **example** data
  - by means of **optimizing** an **objective function.**

When to use ML?

- **Classical algorithm:**



- **Machine learning algorithm**:



When to use ML?

- There is no need to “learn” to calculate payroll.
- Learning is used when
  - **Data** is cheap and abundant;
  - Knowledge is expensive and scarce.

  In particular when
  - Human expertise does not exist (scientific research);
  - Humans are unable to explain their expertise (speech recognition, computer vision);
  - Existing solutions are expensive (routing computer networks).

ML and other disciplines

- **Engineering:**

  Signal processing, optimal control, robotics.
- **Mathematics:**

  Probabilities, statistics, optimization.
- **Computer science:**

  Algorithmics, optimization, databases.
- **Data mining & pattern recognition.**
- **Artificial Intelligence:**

  ML is a **component** of AI.

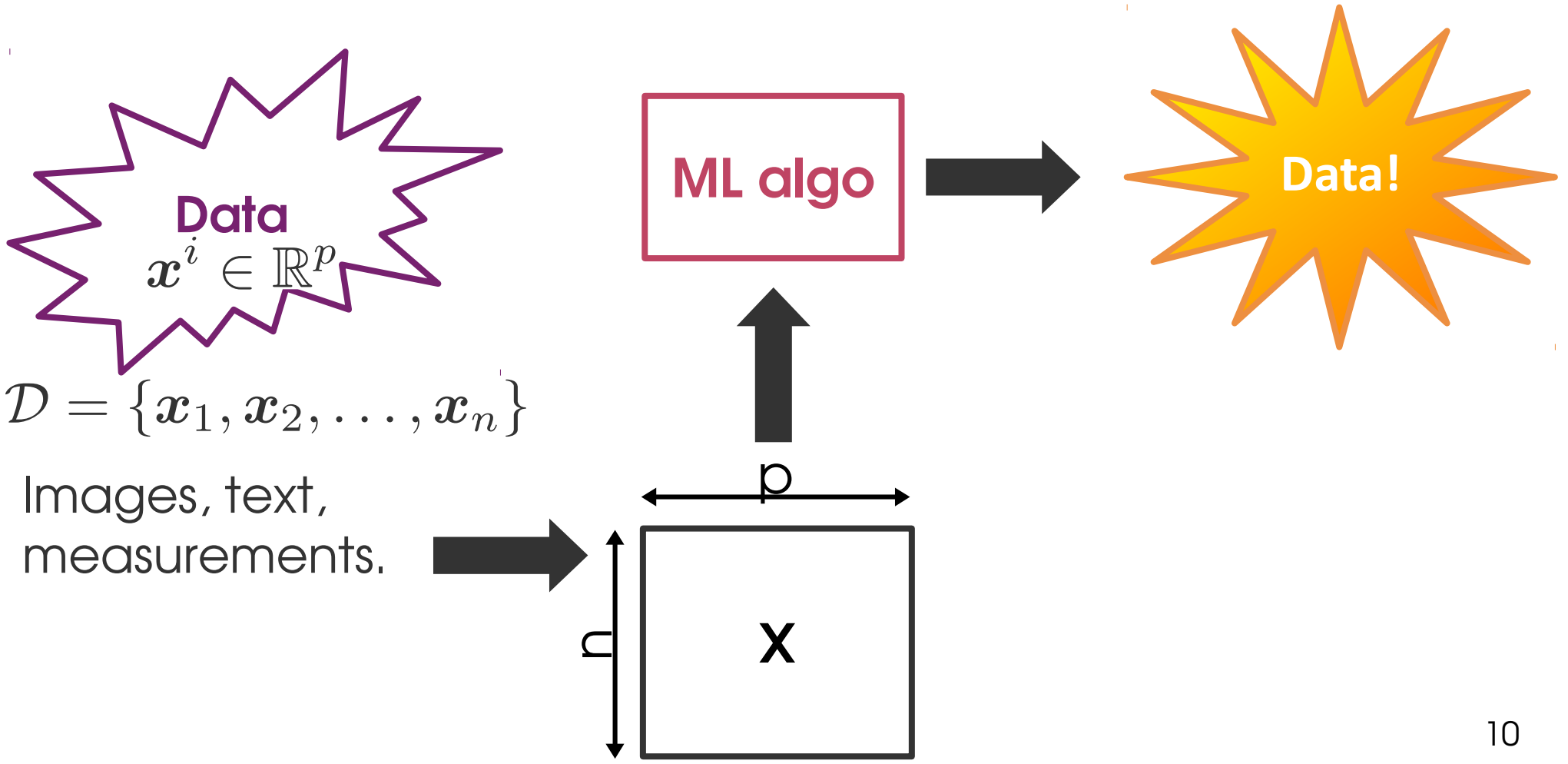
Learning objectives

- **Define machine learning**
- Given a problem
  - **Decide** whether **it can be solved** with machine learning
  - **Decide** as what type of machine learning problem you can formalize it (**unsupervised** – **clustering**, **dimension reduction**, **supervised** – **classification**, **regression**?)
- **Define generalization.**

Zoo of ML Problems

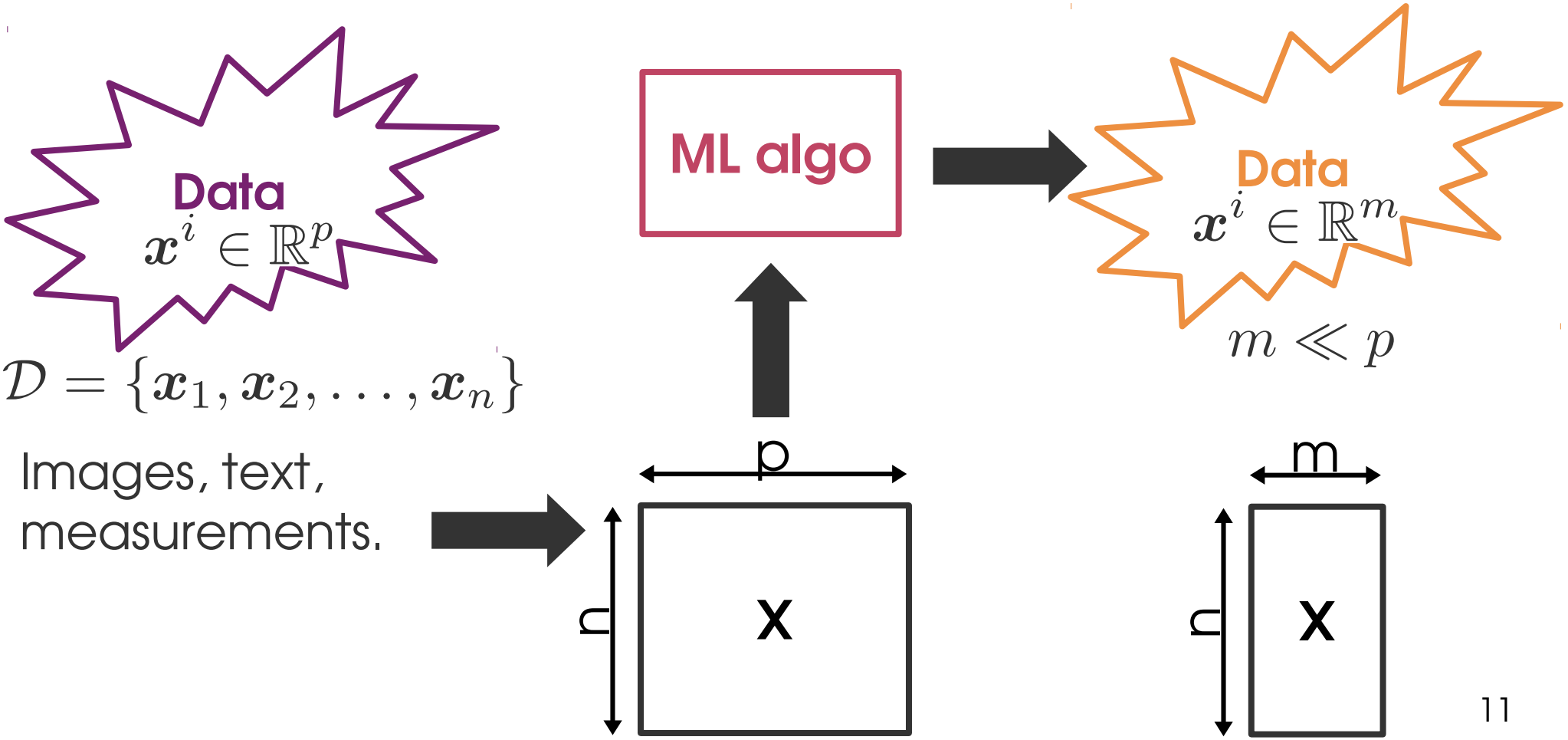
Unsupervised learning

Learn a **new representation** of the data



Dimensionality reduction

Find a **lower-dimensional** representation



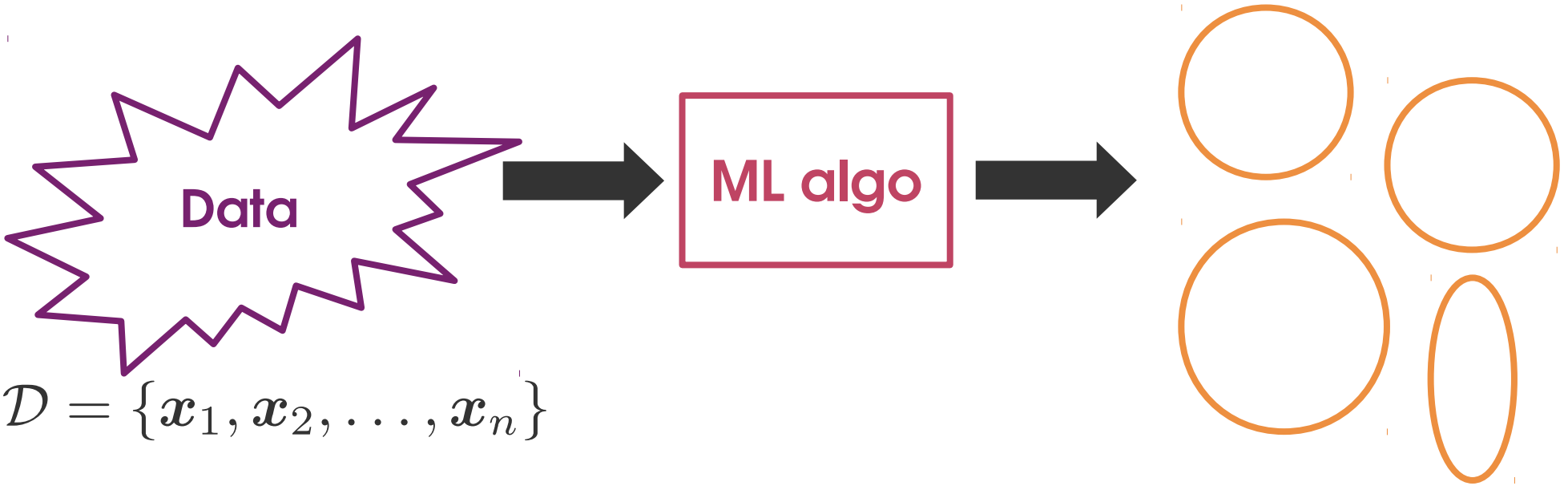
Dimensionality reduction

Find a **lower-dimensional** representation

- Reduce storage **space** & computational **time**
- Remove **redundances**
- **Visualization** (in 2 or 3 dimensions) and **interpretability**.

Clustering

Group **similar** data points together



$\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

- Understand **general characteristics** of the data;
- **Infer some properties** of an object based on how it relates to other objects.

Clustering: applications

- **Customer segmentation**

  Find groups of customers with similar buying behaviors.

- **Topic modeling**

  Groups documents based on the words they contain to identify common topics.

- **Image compression**

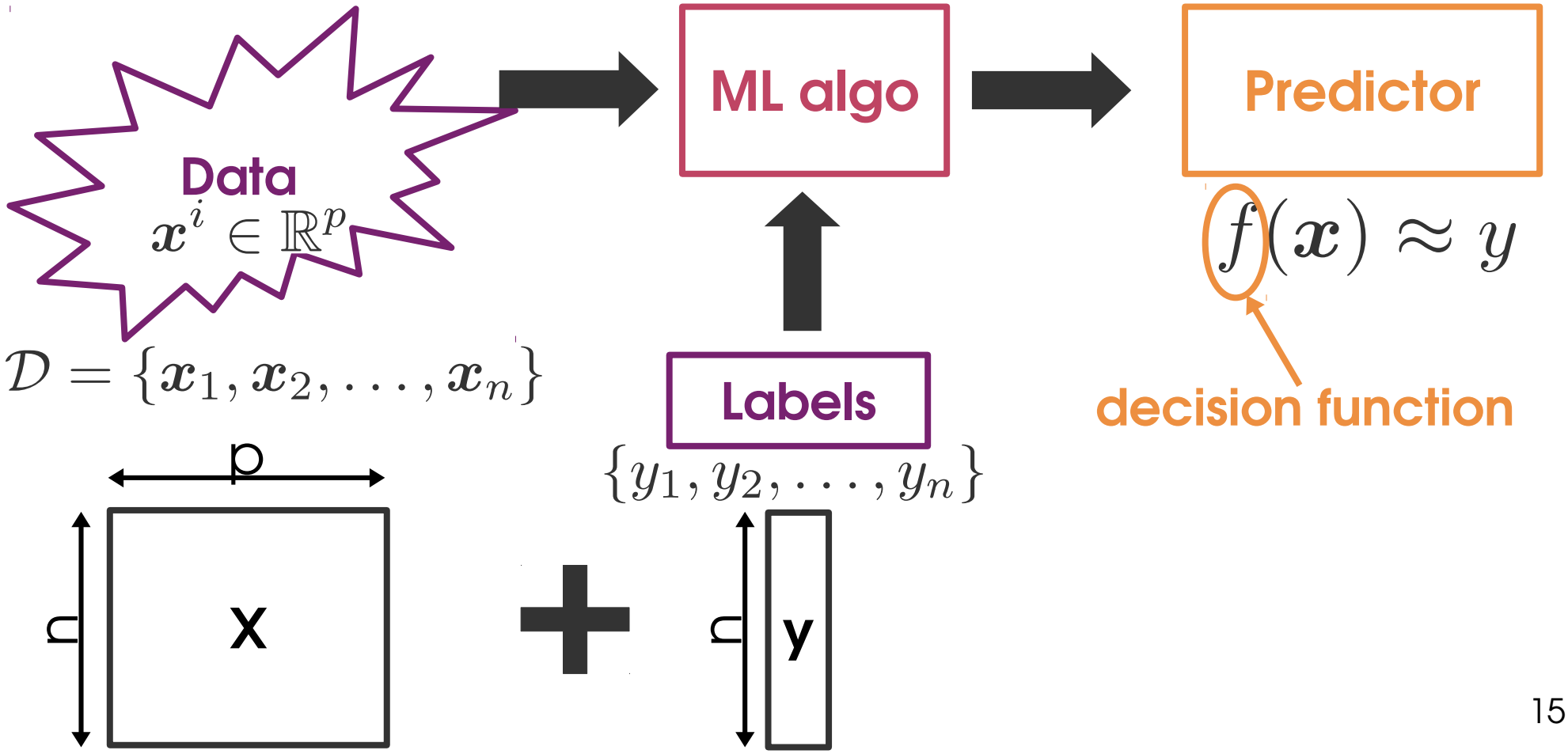
  Find groups of similar pixels that can be easily summarized.

- **Disease subtyping (cancer, mental health)**

  Find groups of patients with similar pathologies (molecular or symptomes level).

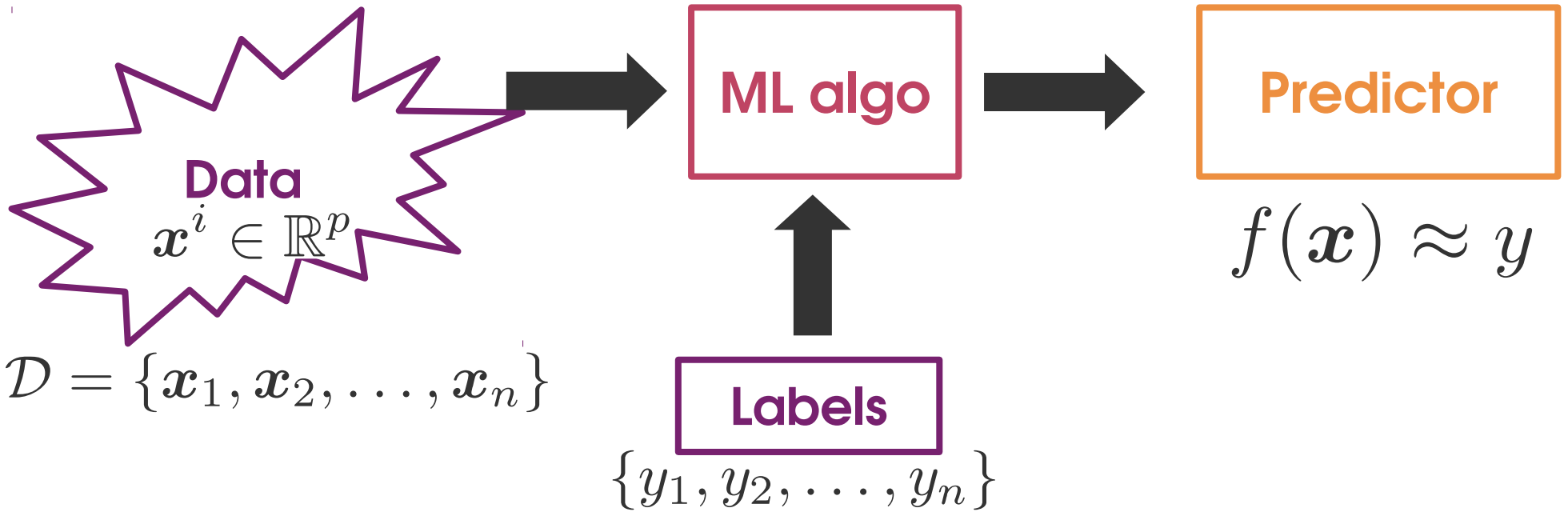
Supervised learning

Make **predictions**



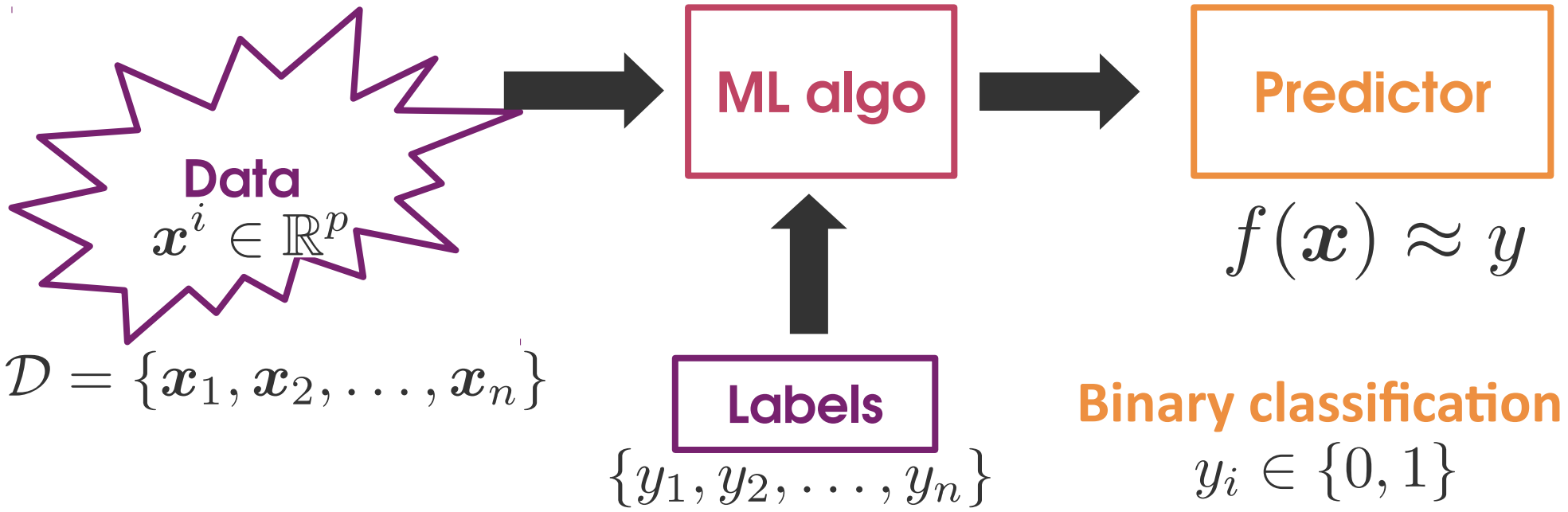
Classification

Make **discrete** predictions



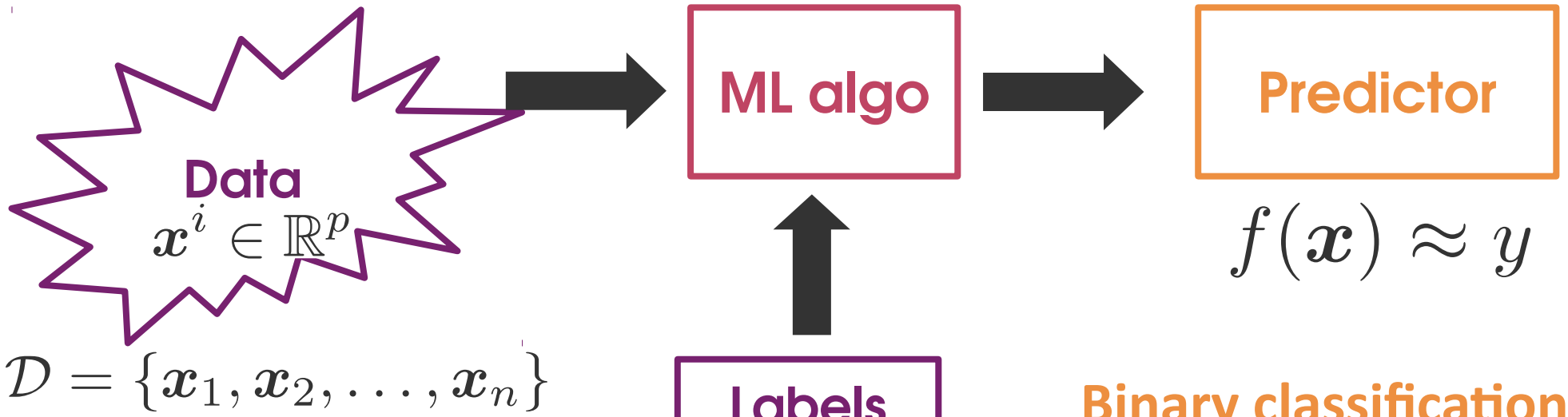
Classification

Make **discrete** predictions



Classification

Make **discrete** predictions



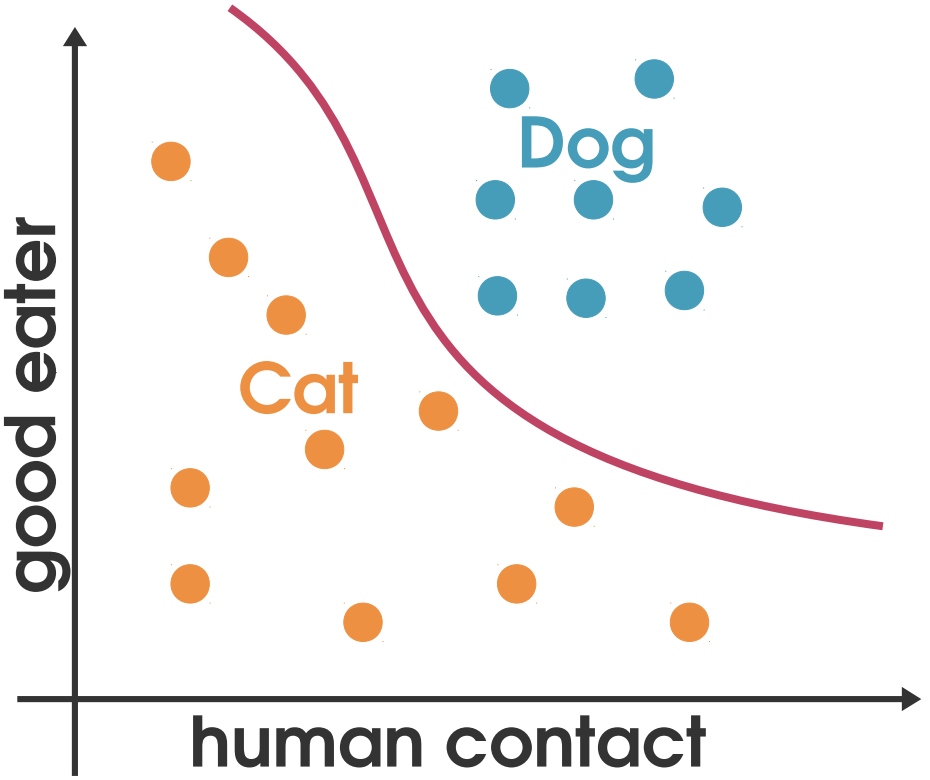
Binary classification

$y_i \in \{0, 1\}$

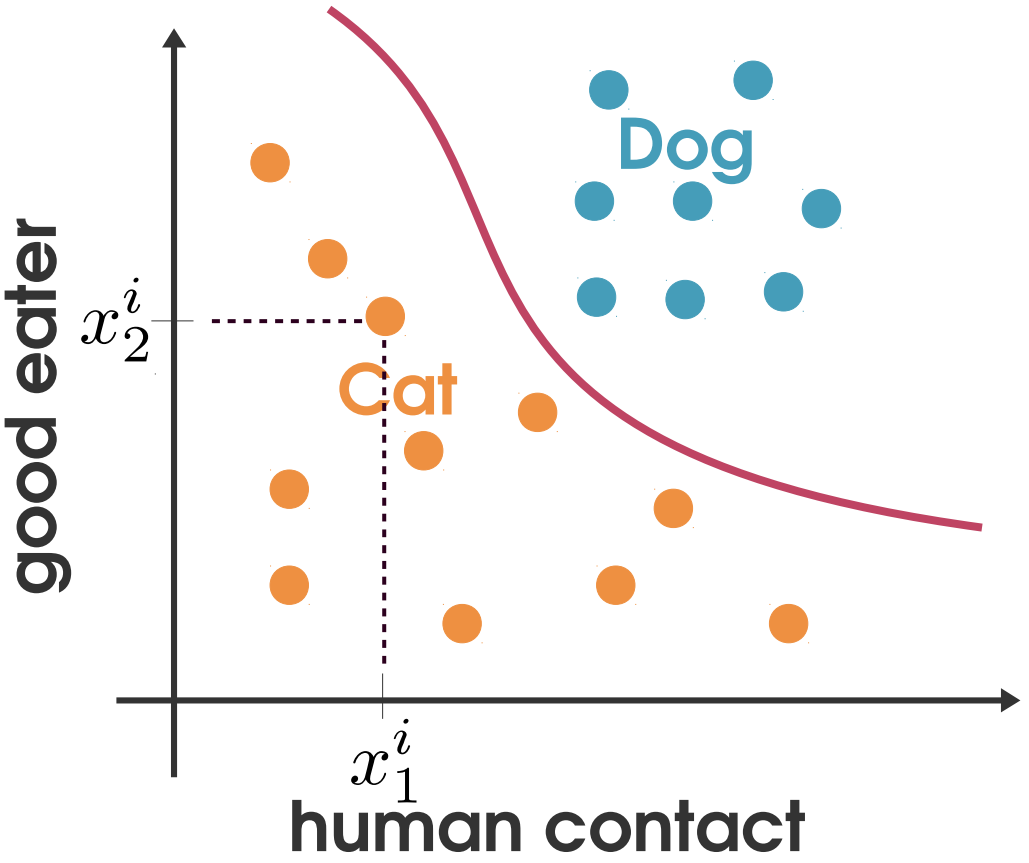
Multi-class classification

$y_i \in \{0, 1, \ldots, k\}$

Classification



Training set $\mathcal{D}$



$$\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$$

$$y^i = \begin{cases} 1 & \text{if } \boldsymbol{x}^i \in \mathcal{P} \\ 0 & \text{if } \boldsymbol{x}^i \in \mathcal{N} \end{cases}$$

$$\boldsymbol{x}^i = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}$$

Given $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$, find f such that $f(\boldsymbol{x}) \approx y$.

Classification: Applications

- **Face recognition**

  Identify faces independently of pose, lighting, occlusion (glasses, beard), make-up, hair style.

- **Vehicle identification** (self-driving cars)

- **Character recognition**

  Read letters or digits independently of different handwriting styles.

- **Sound recognition**

  Which language is spoken? Who wrote this music? What type of bird is this?

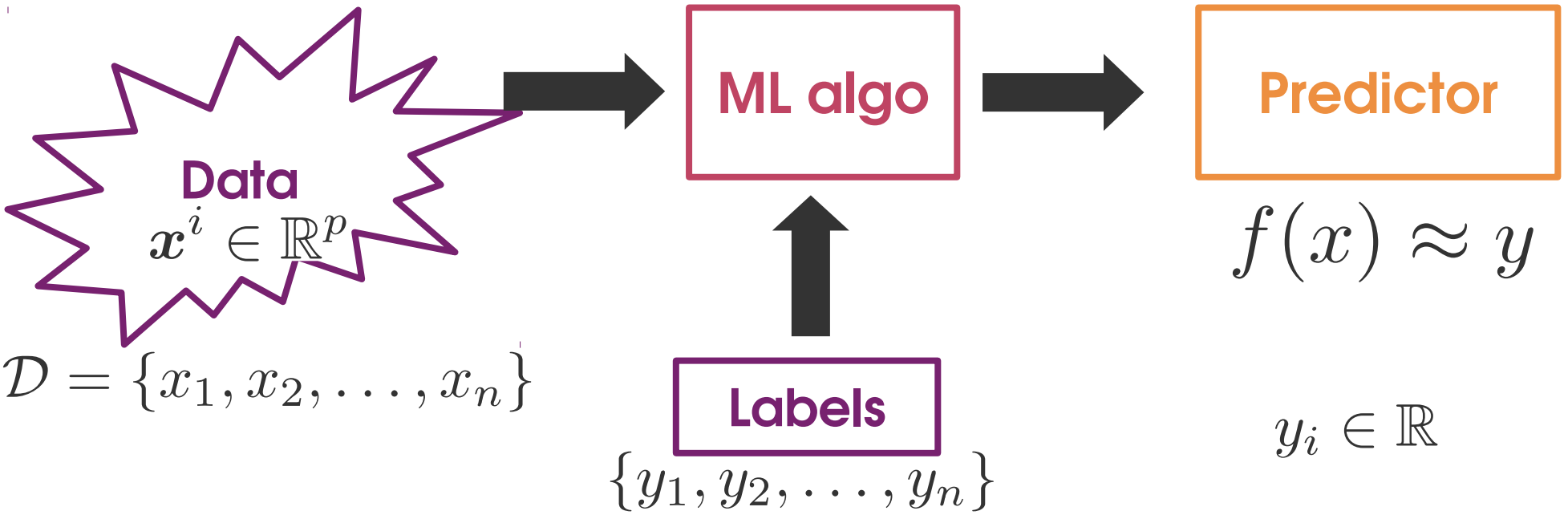
- **Spam detection**

- **Precision medicine**

  Does this sample come from a sick or healthy person? Will this drug work on this patient?

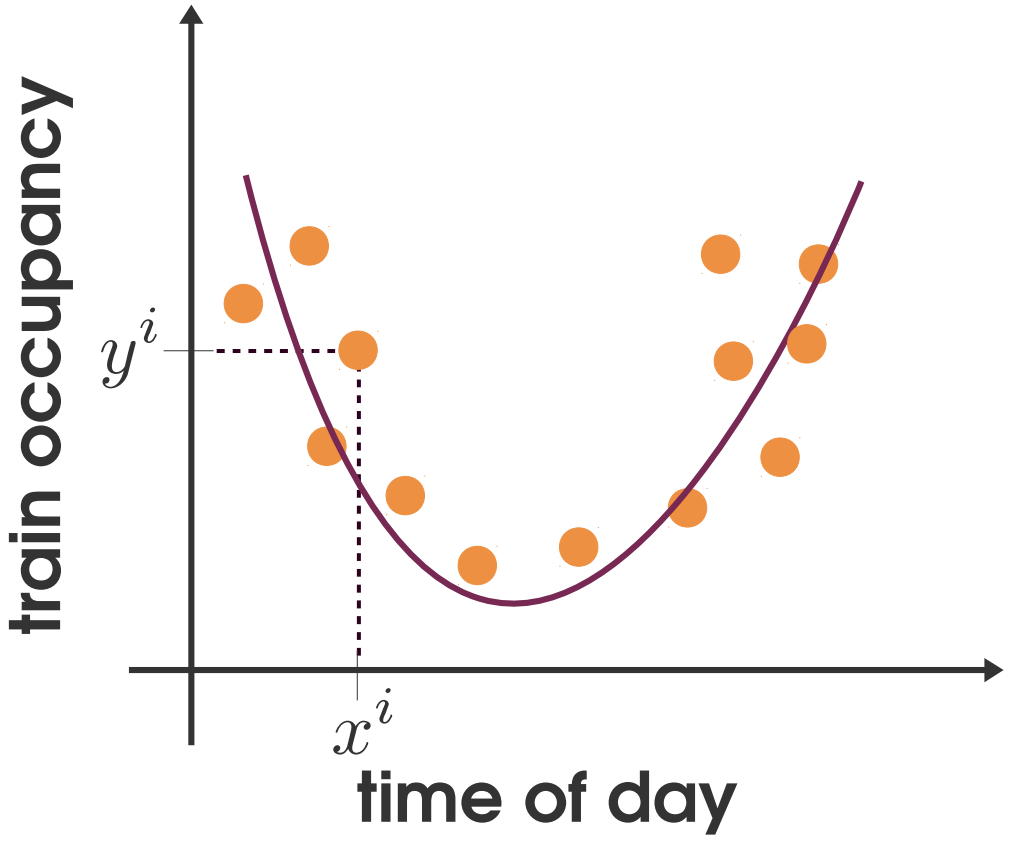
Regression

Make **continuous** predictions



Regression

Regression



$$\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$$

$$y^i \in \mathbb{R}$$

Given $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$, find f such that $f(\boldsymbol{x}) \approx y$.

Regression: Applications

- **Click prediction**

  How many people will click on this ad? Comment on this post? Share this article on social media?

- **Load prediction**

  How many users will my service have at a given time?

- **Algorithmic trading**

  What will the price of this share be?

- **Drug development**

  What is the binding affinity between this drug candidate and its target? What is the sensibility of the tumor to this drug?

Generalization

- **Goal of ML:** *Find a **good** and **useful** approximation of the data.*
- It's easy to build a model that performs well on the training data.
- But how well will it perform on **new data?**
- "Predictions are hard, especially about the future" — Niels Bohr.
- Challenges:
  - **Learn** models that **generalize** well.
  - **Evaluate** whether models generalize well.

Learning objectives

After this course, you should be able to

- **Identify problems** that can be solved by machine learning;
- **Formulate** your problem in machine learning terms
- Given such a problem, **identify** and **apply** the most appropriate classical algorithm(s);
- **Evaluate** and **compare** machine learning algorithms for a particular task.

Course Syllabus

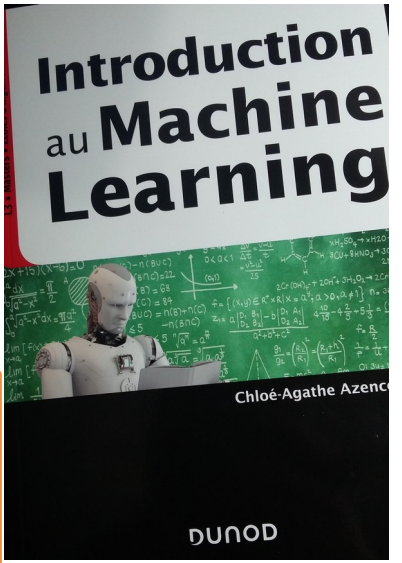
- 1. Introduction
- 2. Linear regression
- 3. Model selection & evaluation
- 4. Regularization
- 5. Dimensionality reduction
- 6. A probabilistic view of ML
- 7. Introduction to artificial neural networks
- 8. Kernel methods
- 9. Nearest neighbors
- 10. Trees and forests
- 11. Clustering

Labs

`https://github.com/chagaz/hpc-ai-ml-2019`

- With **Alexandre Boilley**
- Lab 1: Principal Component Analysis
- Lab 2: Data normalization
- Lab 3: Introduction to the KaggleInClass project
- Lab 4: Linear and logistic regression
- Lab 5: Regularized linear regression
- Lab 6: Nearest neighbors
- Lab 7: Trees and Forests
- Lab 8: Support Vector Machines
- Lab 9: Work on the project

Textbooks



- ***Introduction au Machine Learning***
  Chloé-Agathe Azencott, Dunod
- ***A Course in Machine Learning***
  Hal Daumé III
  `http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf`
- ***The Elements of Statistical Learning***
  Trevor Hastie, Robert Tibshirani and Jerome Friedman
  `http://web.stanford.edu/~hastie/ElemStatLearn/`
- ***Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond***
  Bernhard Schölkopf and Alex Smola
  `http://agbs.kyb.tuebingen.mpg.de/lwk/`
- ***Convex Optimization***
  Stephen Boyd and Lieven Vendenberghe
  `https://web.stanford.edu/~boyd/cvxbook/`

Toolboxes

- **Python: scikit-learn**
  `http://scikit-learn.org`

- **R: Machine Learning Task View**
  `http://cran.r-project.org/web/views/MachineLearning.html`
- **Matlab™: Machine Learning with MATLAB**
  `http://mathworks.com/machine-learning/index.html`
  - Statistics and Machine Learning Toolbox
  - Deep Learning Toolbox
- **Deep learning software**
  - **Theano** `http://deeplearning.net/software/theano/`
  - **TensorFlow** `http://www.tensorflow.org/`
  - **Caffe** `http://caffe.berkeleyvision.org/`
  - **Keras** `https://keras.io/`

Hands-on resources

- *Machine Learning with Scikit-Learn,* Aurélien Géron (fr or en)
- Parcours *Data Scientist* ou *Machine Learning Engineer* sur OpenClassrooms (fr)
- **Python (fr):**
  - Cours *Découvrez les librairies Python pour la data science* sur OpenClassrooms : https://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science
  - Cours *Python pour la programmation scientifique* : http://courspython.com
  - Cours *Python pour un data scientist* (Xavier Dupré) : http://www.xavierdupre.fr/app/ensae_teaching_cs/helpsphinx3/td_2a.html
- https://github.com/Avik-Jain/100-Days-Of-ML-Code
- **Cheat sheets :**
  - Scikit-learn's *choosing the right estimator* : https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html
  - *Which machine learning algorithm should I use ?* https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
  - https://github.com/FavioVazquez/ds-cheatsheets