MA2823: Introduction to Machine Learning

CentraleSupélec — Fall 2017

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://tinyurl.com/ma2823-2017`

  `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**:

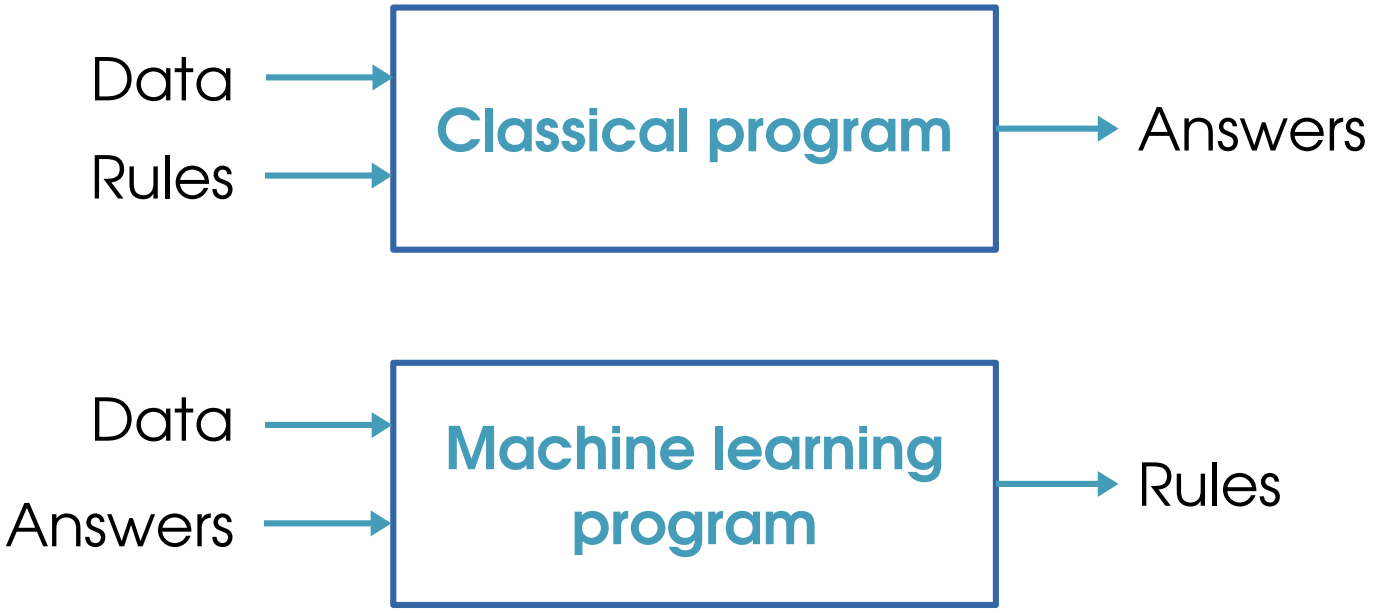
  Modifying a behavior based on experience

  [F. Benureau]

- **Machine learning**: Programming computers to
  - **Model phenomena**
  - by means of **optimizing** an **objective function**
  - using **example** data.

Why Learn?

- There is no need to “learn” to calculate payroll.
- Learning is used when
  - Human expertise does not exist (bioinformatics);
  - Humans are unable to explain their expertise (speech recognition, computer vision);
  - Complex olutions change in time (routing computer networks).



What about AI?

Artificial Intelligence

ML is a subfield of **Artificial Intelligence**

- A system that lives in a **changing environment** must have the ability to **learn** in order to **adapt.**
- ML algorithms are building blocks that make computers behave more intelligently by **generalizing** rather than merely storing and retrieving data (like a database system would do).

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**?)
  - **Describe** it formally in terms of **design matrix**, **features**, **samples**, and possibly **target**.
- **Define** a **loss** function (supervised setting)
- **Define** **generalization.**

What is machine learning?

- **Learning general models from particular examples** (data)
  - **Data** is (mostly) cheap and abundant;
  - **Knowledge** is expensive and scarce.
- Example in retail:

  From customer transactions to consumer behavior

  People who bought “Game of Thrones” also bought “Lord of the Rings” [amazon.com]
- Goal: **Build a model that is a good and useful approximation to the data.**

What is machine learning?

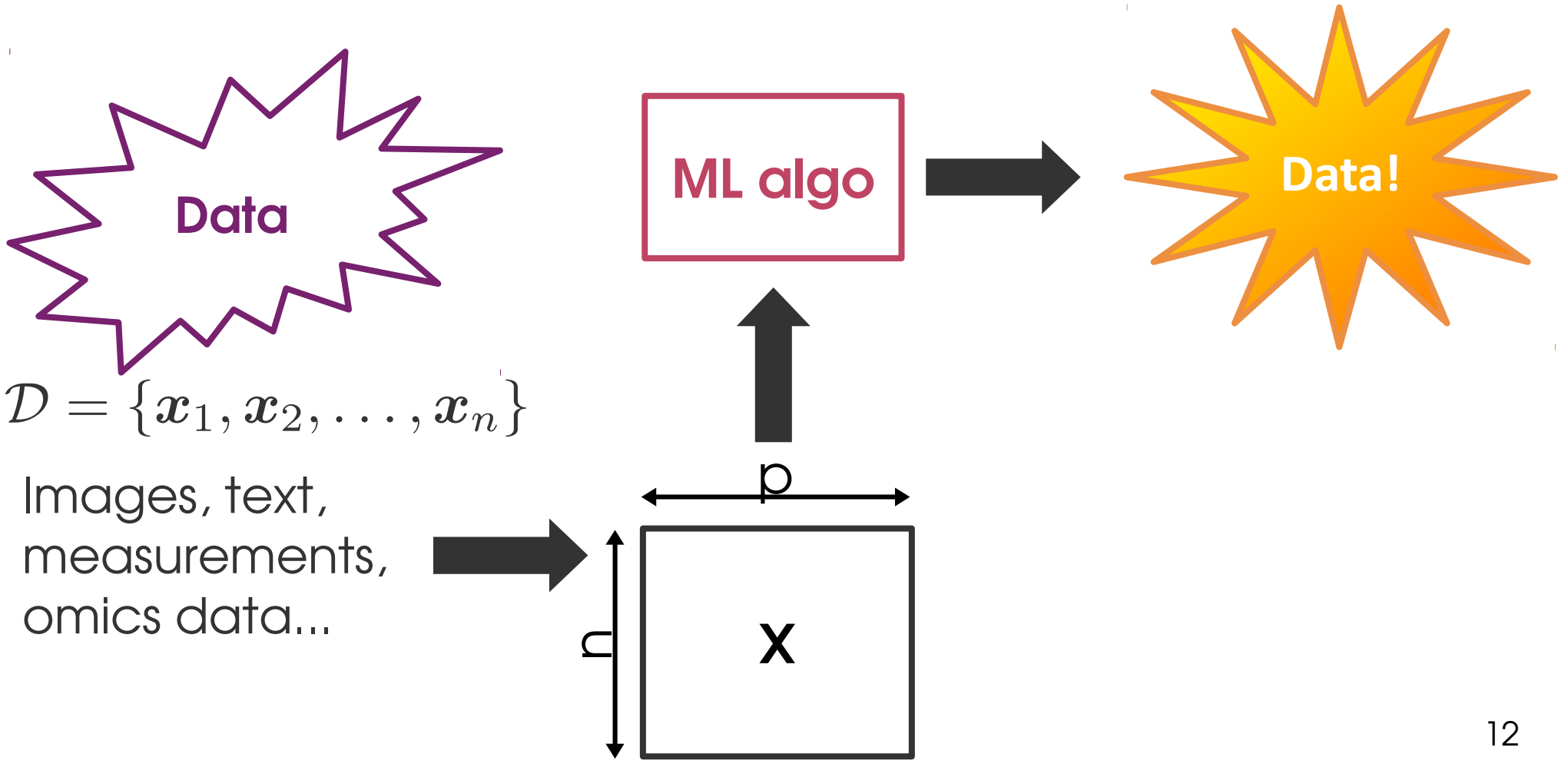
- **Optimizing a performance criterion using example data or past experience.**
- Role of **Statistics:**

  Build mathematical models to make inference from a sample.
- Role of **Computer Science:** Efficient algorithms to
  - Solve the optimization problem;
  - Represent and evaluate the model for inference.

Zoo of ML Problems

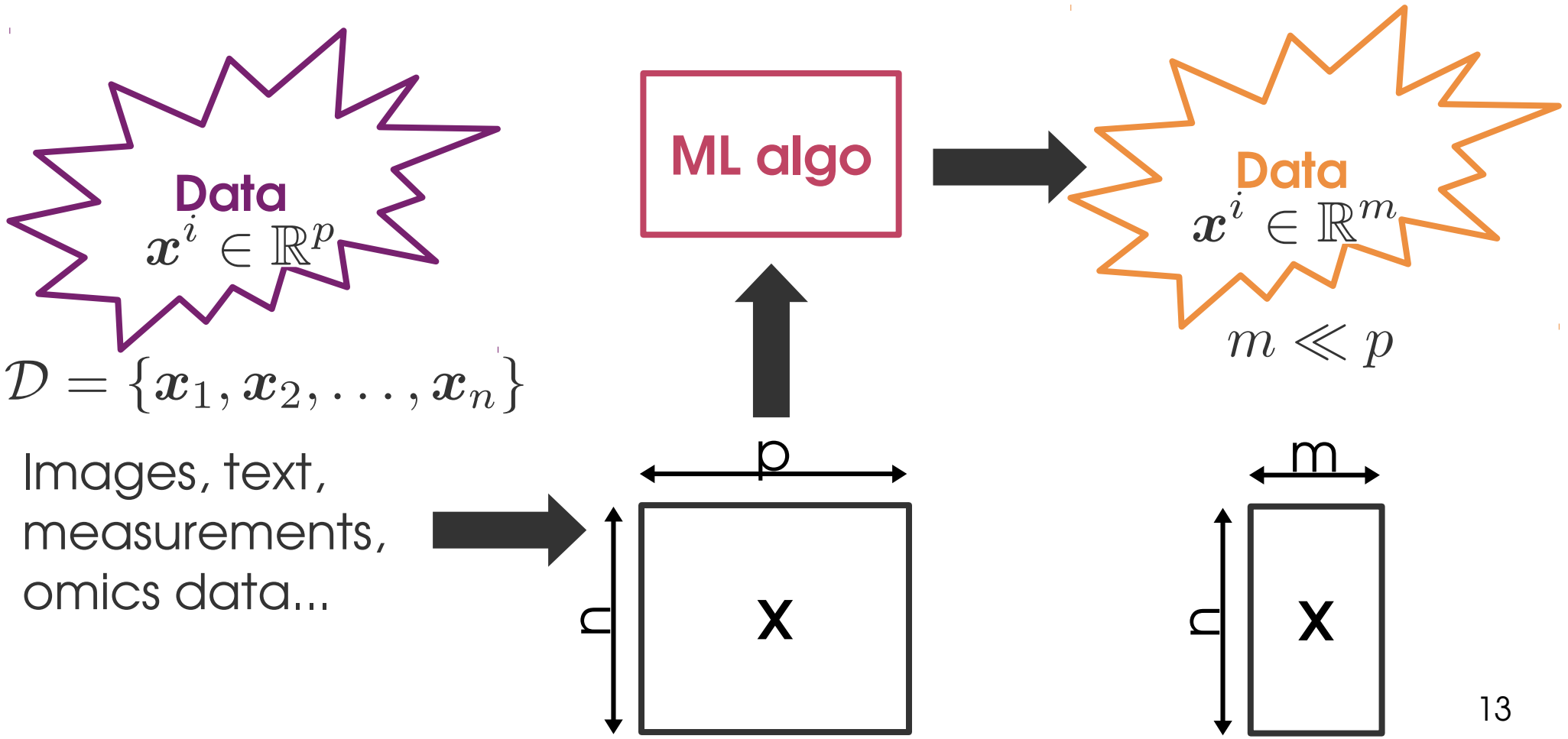
Unsupervised learning

Learn a **new representation** of the data



Dimensionality reduction

Find a **lower-dimensional** representation



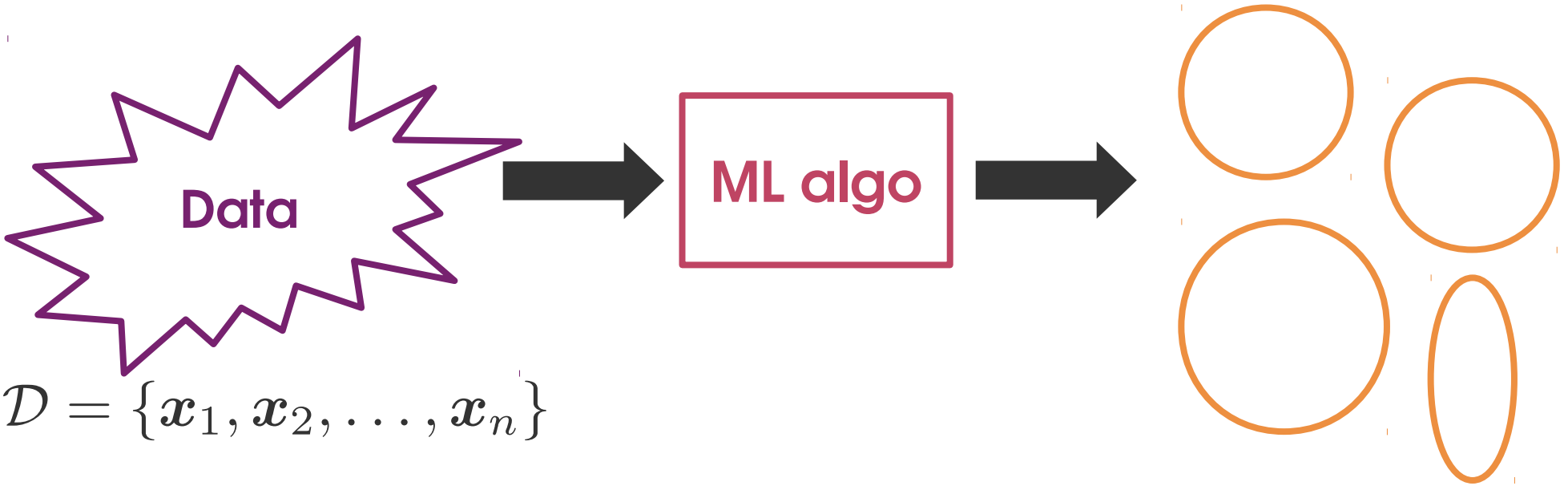
Dimensionality reduction

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**

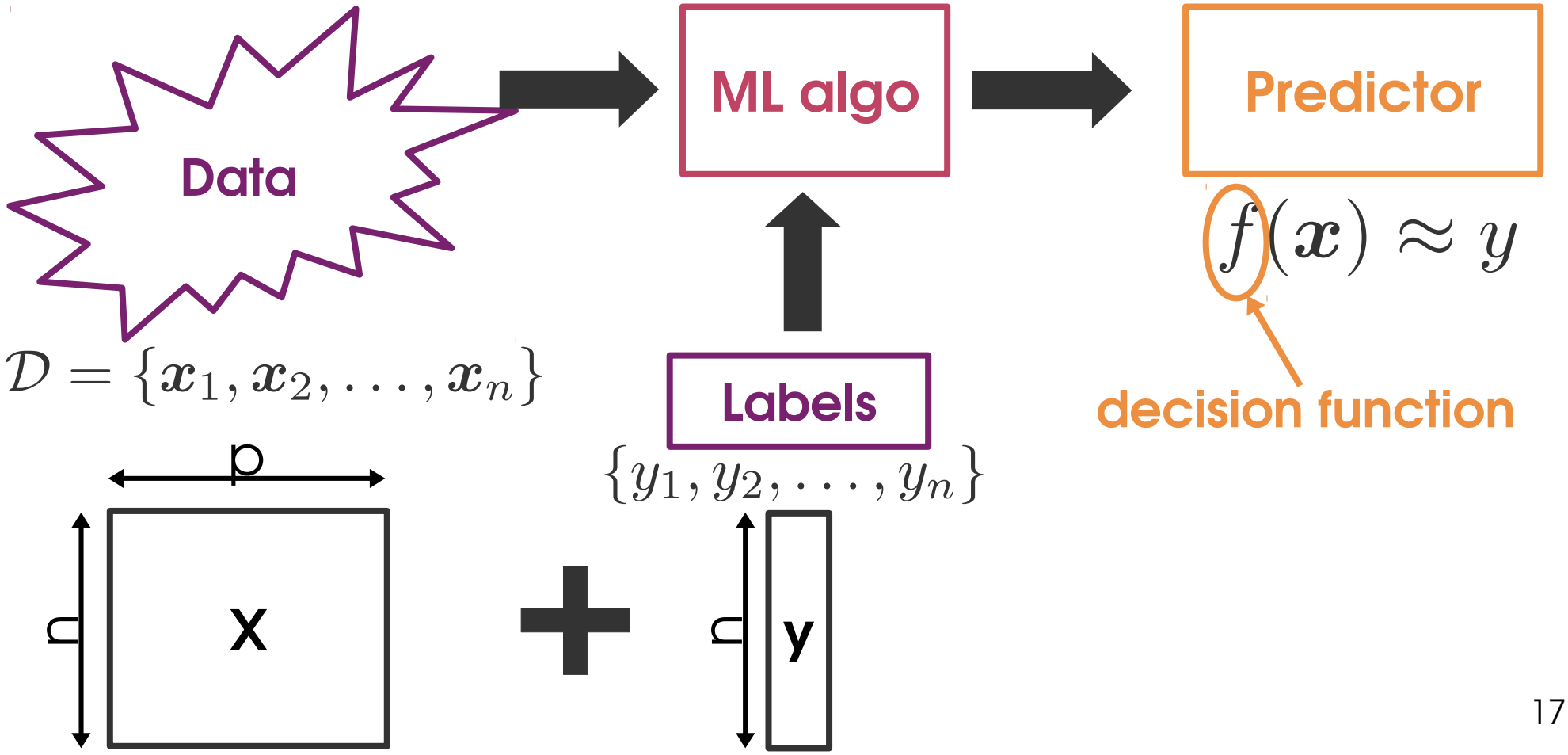
  Find groups of similar pixels that can be easily summarized.

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

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

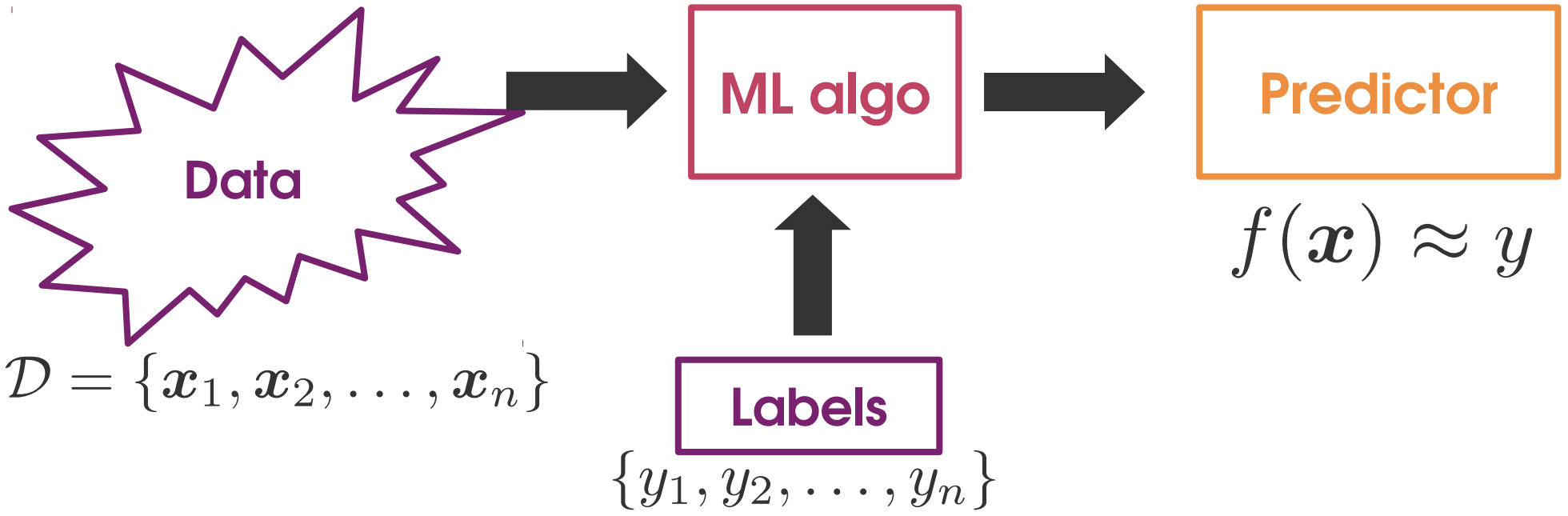
Supervised learning

Make **predictions**



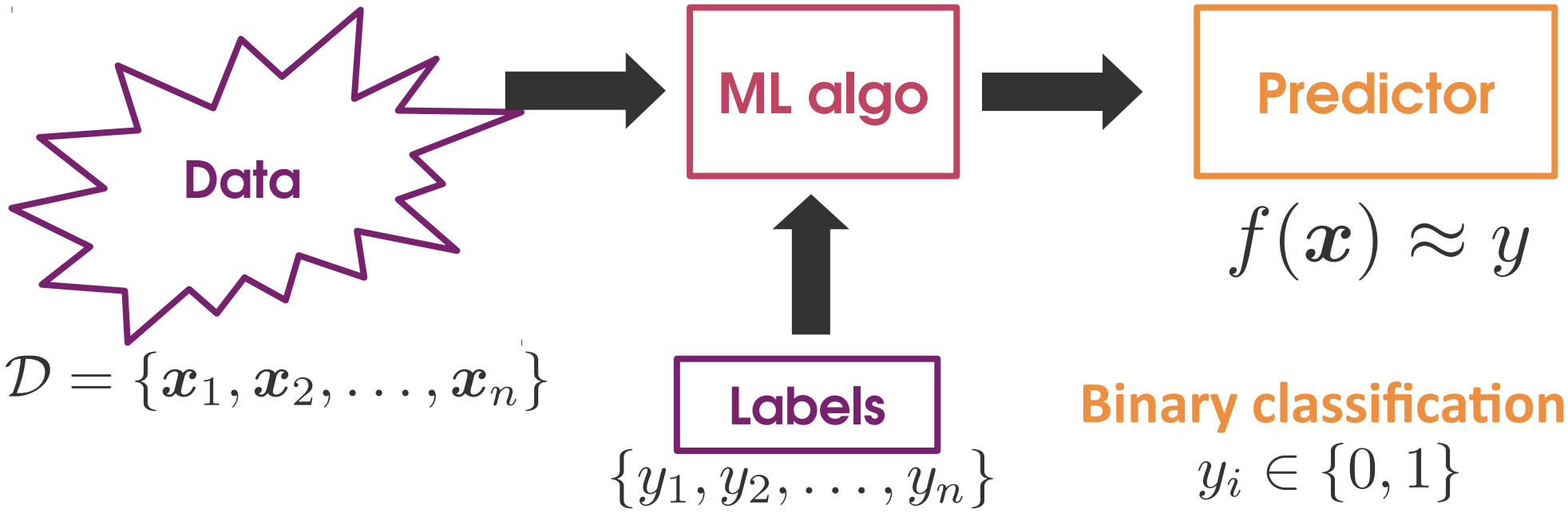
Classification

Make **discrete** predictions



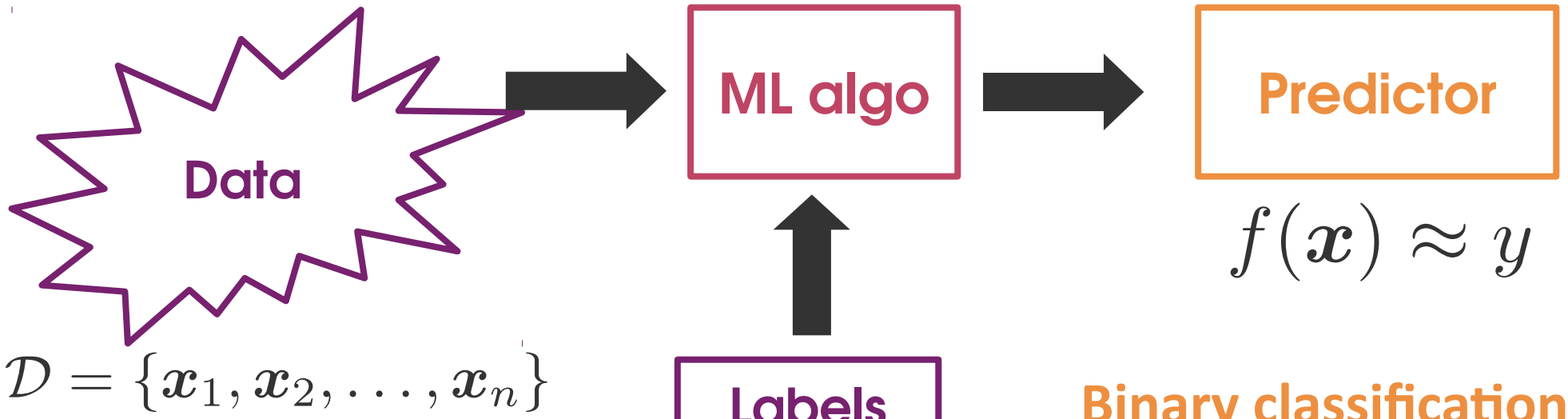
Classification

Make **discrete** predictions



Classification

Make **discrete** predictions



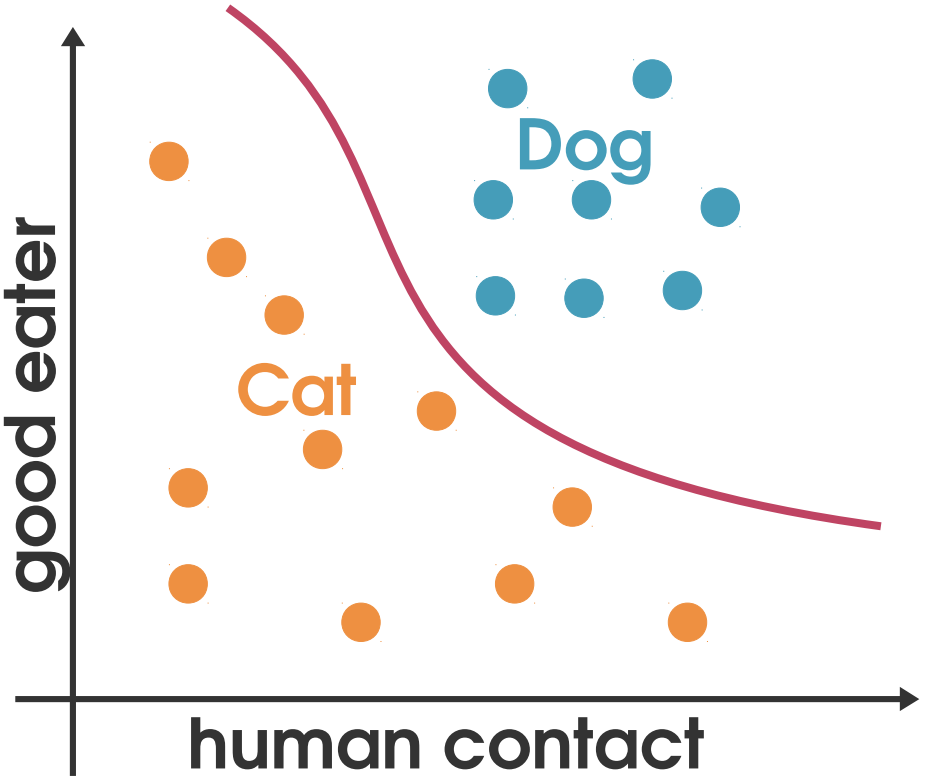
Binary classification

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

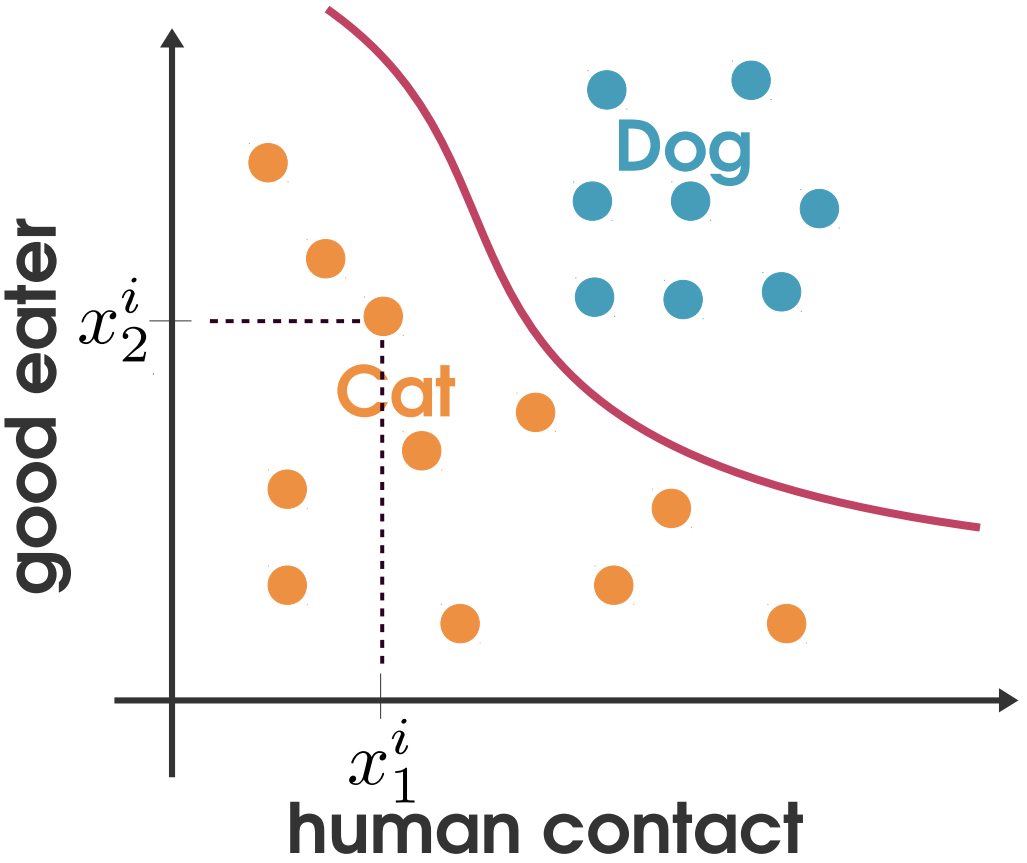
Multi-class classification

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

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?

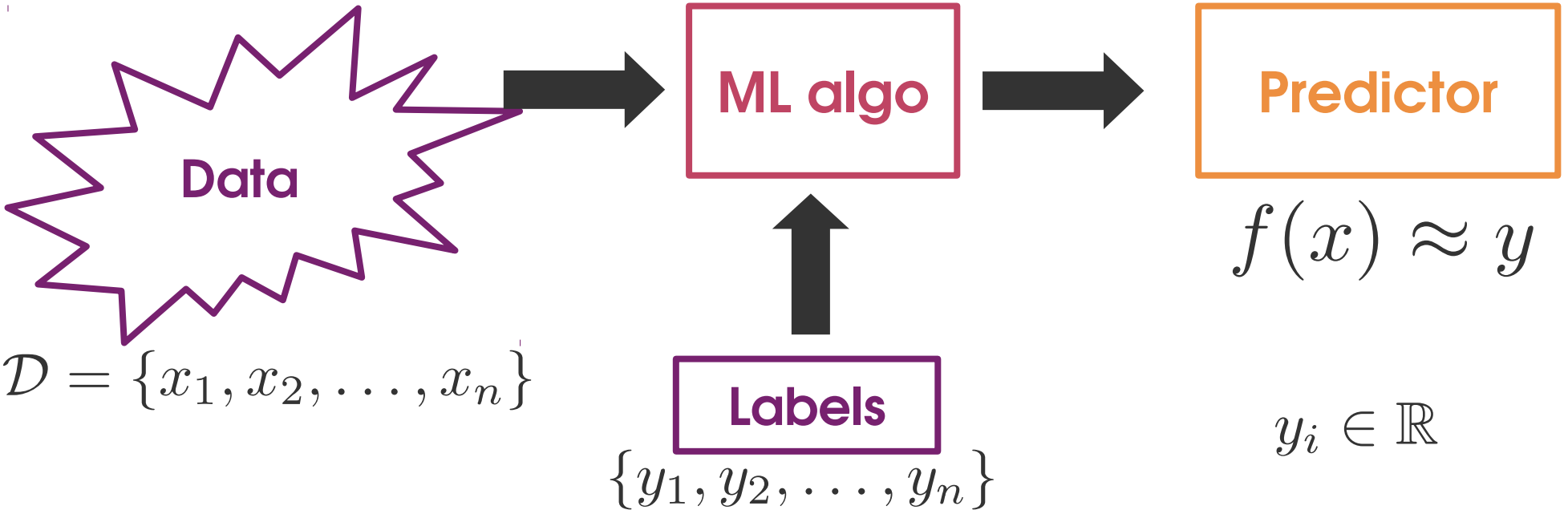
- **Spam detection**

- **Precision medicine**

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

Regression

Make **continuous** predictions



$\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$

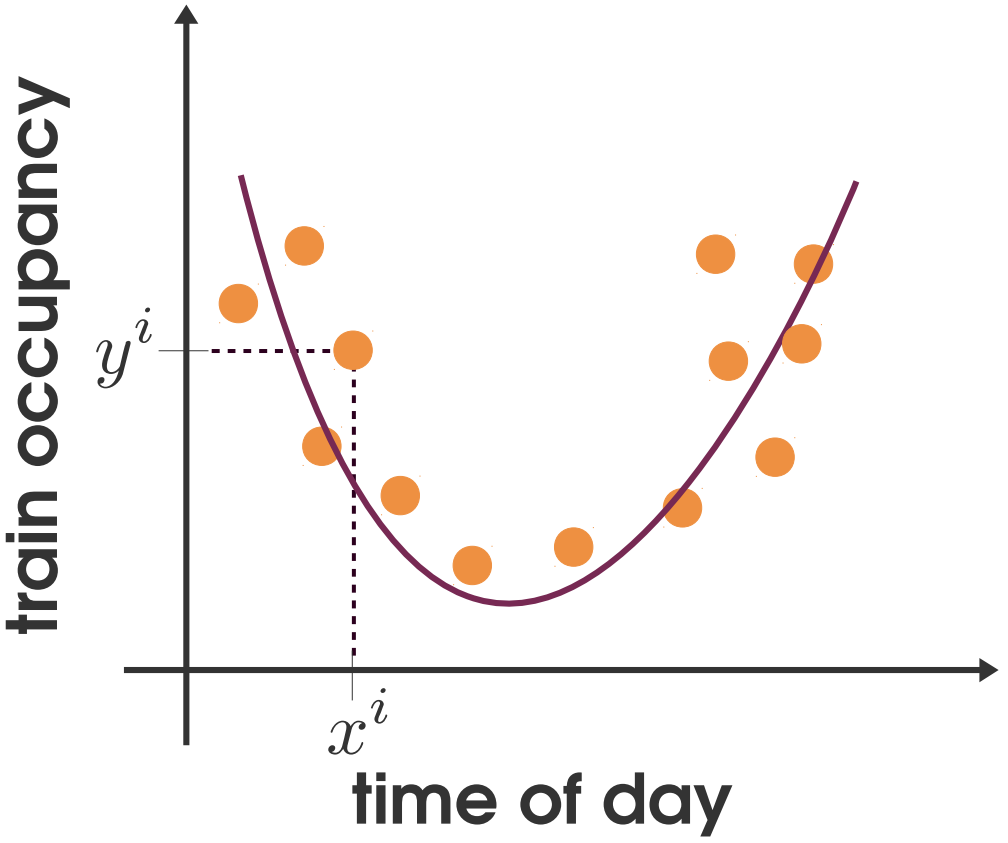
$\{y_1, y_2, \ldots, y_n\}$

$f(x) \approx y$

$y_i \in \mathbb{R}$

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**

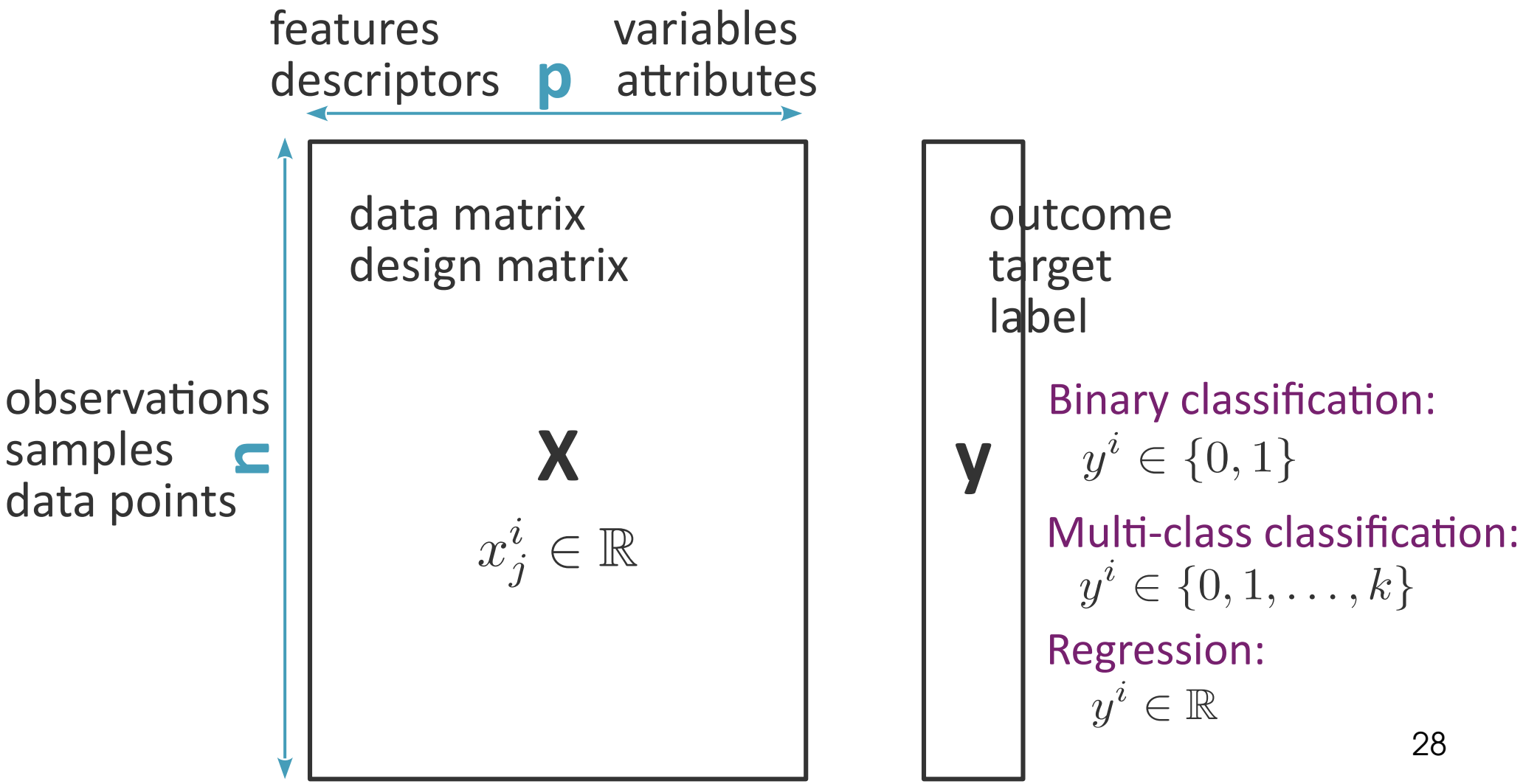
  What will the price of this share be?

- **Drug development**

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

Supervised learning setting

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

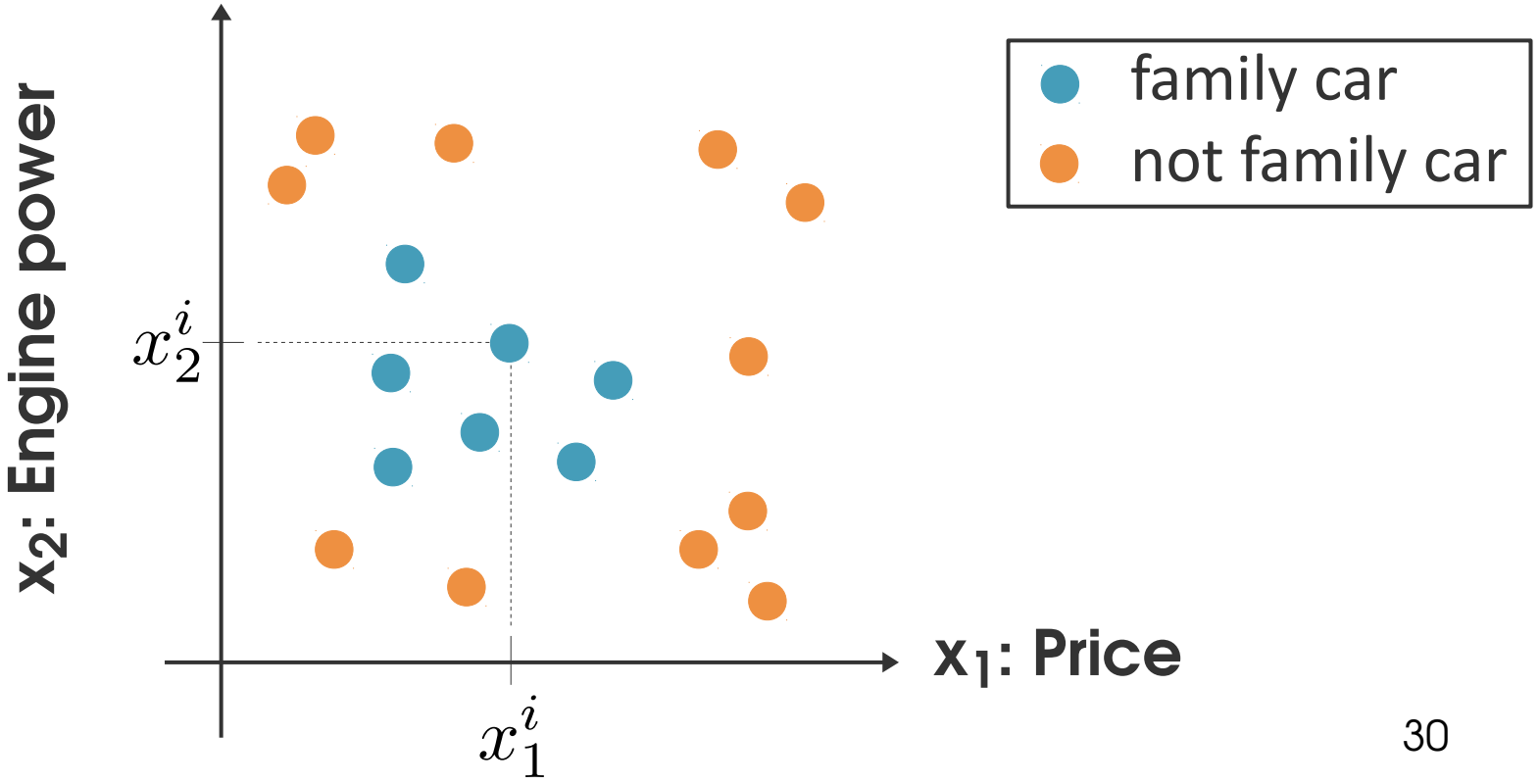


Hypothesis class

- **Hypothesis class** $\mathcal{F}$
  - The **space of possible decision functions** we are considering
  - Chosen based on our **beliefs** about the problem

Hypothesis class

- **Hypothesis class** $\mathcal{F}$
  - The **space of possible decision functions** we are considering
  - Chosen based on our **beliefs** about the problem



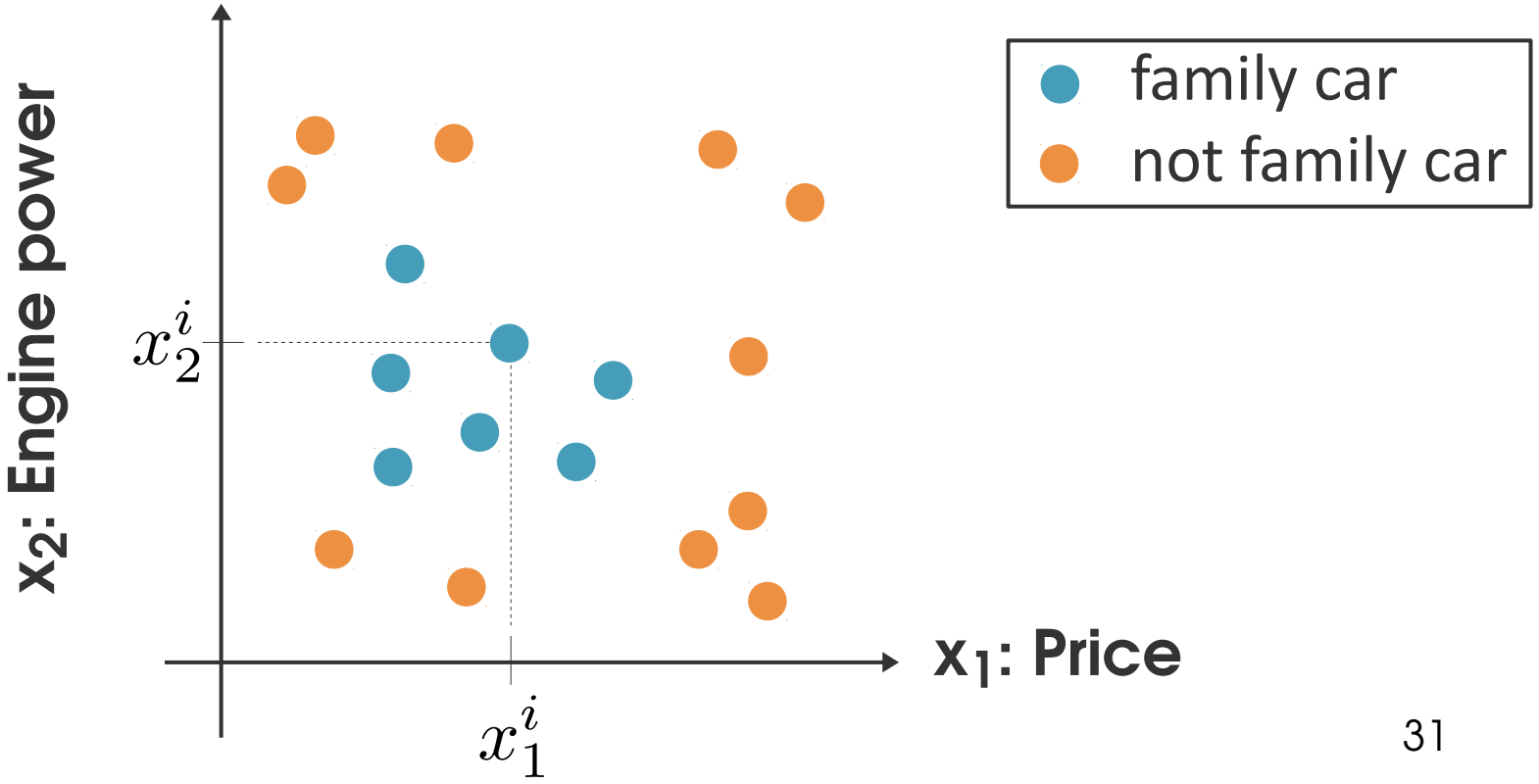
Hypothesis class

- **Hypothesis class** $\mathcal{F}$

  The space of possible decision functions we are

  chosen based on our beliefs about the problem

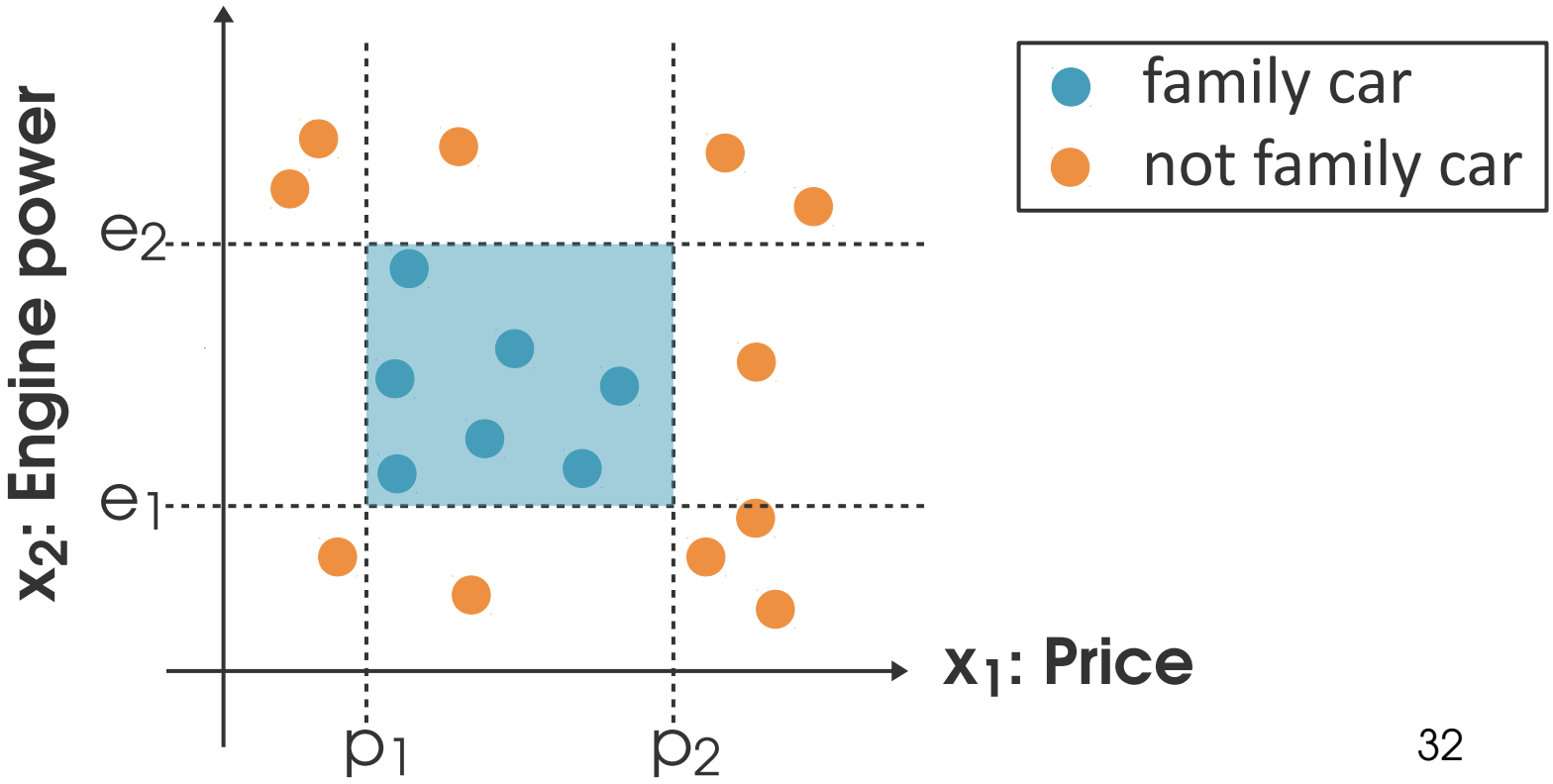
What shape do you think the discriminant should take?



Hypothesis class

- **Hypothesis class** $\mathcal{F}$
  - Belief: the decision function is a rectangle

$$(p_1 \leq x_1 \leq p_2) \text{ AND } (e_1 \leq x_2 \leq e_2)$$



Loss function

- **Loss function** (or **cost function**, or **risk**):

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

$$y, f(\boldsymbol{x}) \mapsto \mathcal{L}(y, f(\boldsymbol{x}))$$

  Quantifies how far the decision function is from the truth (= **oracle**).

- E.g.
  - $\mathcal{Y} = \{0, 1\}$ $\quad \mathcal{L}(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if } y = f(\boldsymbol{x}) \\ 1 & \text{otherwise} \end{cases}$

Loss function

- **Loss function** (or **cost function**, or **risk**):

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

$$y, f(\boldsymbol{x}) \mapsto \mathcal{L}(y, f(\boldsymbol{x}))$$

  Quantifies how far the decision function is from the truth (= **oracle**).

- E.g.
  - $\mathcal{Y} = \mathbb{R} \quad \mathcal{L}(y, f(\boldsymbol{x})) = ||y - f(\boldsymbol{x})||^2$

Loss function

- **Loss function** (or **cost function**, or **risk**):

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

$$y, f(\boldsymbol{x}) \mapsto \mathcal{L}(y, f(\boldsymbol{x}))$$

  Quantifies how far the decision function is from the truth (= **oracle**).

- **Empirical risk** on dataset $\mathcal{D}$

$$E_{\mathcal{D}} = \sum_{i=1}^{n} \mathcal{L}(y^i, f(\boldsymbol{x}^i))$$

Supervised learning: 3 ingredients

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

A good and useful approximation

- Chose a **hypothesis class** $\mathcal{F}$
  - Parametric methods — e.g. $f(x) = \sum_{j=1}^{p} \beta_j x_j$
  - Non-parametric methods — e.g. f(x) is the label of the point closest to x.
- Chose a **loss function** $\mathcal{L}$

  Empirical error: $E_{\mathcal{D}} = \sum_{i=1}^{n} \mathcal{L}(y^i, f(\boldsymbol{x}^i))$
- Chose an **optimization procedure**

$$f^* = \arg\min_{f \in \mathcal{F}} E_{\mathcal{D}}$$

Generalization

A good and useful approximation

- 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.
  - Learn models that **generalize** well.
  - Evaluate whether models generalize well.

Artificial intelligence

Electrical engineering

Signal processing

Pattern recognition

Engineering

Optimization

Knowledge discovery
in databases

Computer science

Data mining

Inference

Big data

Business

Discriminant analysis

Statistics

Data science

Induction

http://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html

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);
- **Implement** some of these algorithms yourself;
- **Evaluate** and **compare** machine learning algorithms for a particular task.

Course Syllabus

- Sep 29
  - 1. Introduction
  - **2. Convex optimization**
- Oct 2
  - **3. Dimensionality reduction**
  - Lab: Principal component analysis + Jupyter, pandas, and scikit-learn.
- Oct 6
  - **4. Model selection**
  - Lab: Convex optimization with scipy.optimize
- Oct 13
  - **5. Bayesian decision theory**
  - Lab: Intro to Kaggle challenge
- Oct 20
  - **6. Linear regression**
  - Lab: Linear regression

- Nov 10
  - **7. Regularized linear regression**
  - Lab: Regularized linear regression
- Nov 17
  - **8. Nearest-neighbor approaches**
  - Lab: Nearest-neighbor approaches
- Nov 24
  - **9. Tree-based approaches**
  - Lab: Tree-based approaches
- Dec 01
  - **10. Support vector machines**
  - Lab: Support vector machines
- Dec 08
  - **11. Neural networks**
  - Deep learning (Joseph Boyd) + Bioimage informatics applications (Peter Naylor)
- Dec 15
  - **12. Clustering**
  - Lab: Clustering

Labs

Bring your laptop!

There will be **power plugs** and **wifi.**

Instructions on how to set up your computer:

`https://github.com/chagaz/ma2823_2017`

and in the syllabus

- **TAs:**
  - **Josehp Boyd** `joseph.boyd@mines-paristech.fr`
  - **Benoît Playe** `benoit.playe@mines-paristech.fr`
  - **Mihir Sahasrabudhe** `mihir.sahasrabudhe@centralesupelec.fr`

kaggle challenge project

How Many Shares? Challenge

https://www.kaggle.com/c/how-many-shares

- **Predict the number of shares on social media** for articles from the same media site
  - Regression
  - From article length, topics, subjectivity and much more.
- **Evaluation on**
  - Insights learned
  - Prediction performance.

Evaluation

- **Final exam (60 pts)** **December 22, 2016**
  - Pen and paper
  - Closed book
- **Kaggle project (30 pts)** **December 22, 2016**
  - Written report (25 pts)
  - Position in the leaderboard (5pts)
  - Introduction: October 13, 2017
- **Homework (10 pts)** **1 assignment each week**
  - To get the points: turn it in!

Homework

- **One assignment per week**
  - Similar to the questions you'll be asked at the exam
  - Turn it in **online**

    `http://tinyurl.com/ma2823-2017-hw`
  - **Solution will be posted the day after the due date**
  - Worth 1pt if you turn it in.

Resources

- **Course website**

  `http://tinyurl.com/ma2823-2017`

  - **Syllabus**
  - **2 days before the lecture:** printable lecture handout
  - **Shortly after the lecture:**
    - HW Problem n+1
    - Lecture slides
    - HW Solution n.

- **GitHub repository (labs)**

  `https://github.com/chagaz/ma2823_2017`

Textbooks

- ***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/`

Resources: Datasets

- **UCI Repository:**
  http://www.ics.uci.edu/~mlearn/MLRepository.html
- **KDnuggets Datasets:**
  http://www.kdnuggets.com/datasets/index.html
- **ImageNet:** http://www.image-net.org/
- **Enron Email Dataset:** http://www.cs.cmu.edu/~enron/
- **Million Song Dataset:**
  http://labrosa.ee.columbia.edu/millionsong/
- **IMDB Data:** http://www.imdb.com/interfaces
- **Données publiques françaises:** https://www.data.gouv.fr/
- **TunedIT:** http://www.tunedit.org/
- **Knoema:** https://knoema.com/

Resources: Journals

- **Journal of Machine Learning Research**
  http://jmlr.csail.mit.edu/
- **IEEE Transactions on Pattern Analysis and Machine Intelligence**
  https://www.computer.org/portal/web/tpami
- **Annals of Statistics** http://imstat.org/aos/
- **Journal of the American Statistical Association**
  http://www.tandfonline.com/toc/uasa20/current
- **Machine Learning** http://link.springer.com/journal/10994
- **Neural Computation** http://www.mitpressjournals.org/loi/neco
- **Neural Networks**
  http://www.journals.elsevier.com/neural-networks
- **IEEE Transactions on Neural Networks and Learning Systems**
  http://cis.ieee.org/ieee-transactions-on-neural-networks-and-learning-systems.html

Resources: Conferences

- International Conference on Machine Learning (ICML)
  http://www.icml.cc/
- Neural Information Processing Systems (NIPS) http://www.nips.cc/
- International Conference on Learning Representations (ICLR)
  http://www.iclr.cc/
- European Conference on Machine Learning (ECML)
  http://www.ecmlpkdd.org/
- International Conference on AI & Statistics (AISTATS)
  http://www.aistats.org/
- Uncertainty in Artificial Intelligence (UAI) http://www.auai.org/
- Computational Learning Theory (COLT)
  http://www.learningtheory.org/past-conferences-2/
- Knowledge Discovery and Data Mining (KDD) http://www.kdd.org/
- International Conference on Pattern Recognition (ICPR)
  http://www.icpr2017.org/