# Lecture 1: Introduction

## Machine Learning 2025

Federico Chiariotti (federico.chiariotti@unipd.it)

Università degli Studi di Padova

## Part 1:
## Basic course information

Università
degli Studi
di Padova

## The course

- ➔ Machine learning (SCP8082660)
- ➔ ICT for Internet and multimedia: INP9087775

If you are from other courses please notify me

This course is worth 6 ECTS (48 hours)

We will have 15 lectures and 9 labs

## Lectures

- ➔ Tuesday, 16:30 – LabP104 (DFA: via Belzoni, 7)
- ➔ Friday, 14:30 – Room C (DFA: via Paolotti, 9)

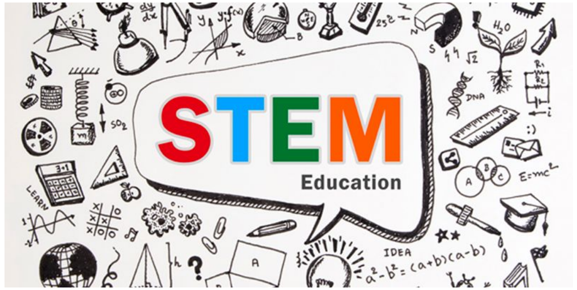
Would you prefer a break or would you rather finish early?

Recordings will be available **for 2 weeks only**

## Elearning

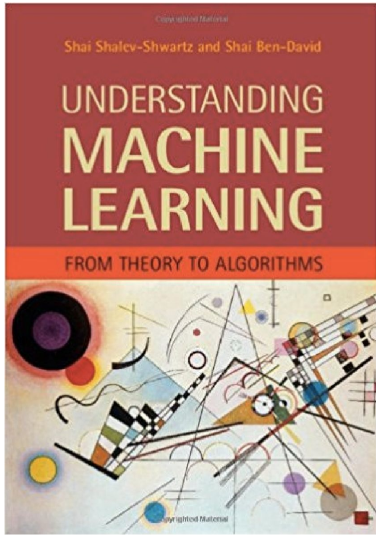
All course materials and announcements will be published through the course Elearning page. The password for manual enrollment is **ML2526_pw** – but most of you should be enrolled automatically.

https://stem.elearning.unipd.it



## Course material

- ➔ Main book: Shalev-Shwartz, Shai, & Ben-David, Shai, *Understanding machine learning: From theory to algorithms*, Cambridge University Press, 2014
- ➔ PDF available from the authors at http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html
- ➔ Lecture slides: available on Elearning
- ➔ Lab solutions and notebooks: available on Elearning


Shai Shalev-Shwartz and Shai Ben-David
UNDERSTANDING MACHINE LEARNING
FROM THEORY TO ALGORITHMS

## Exam format

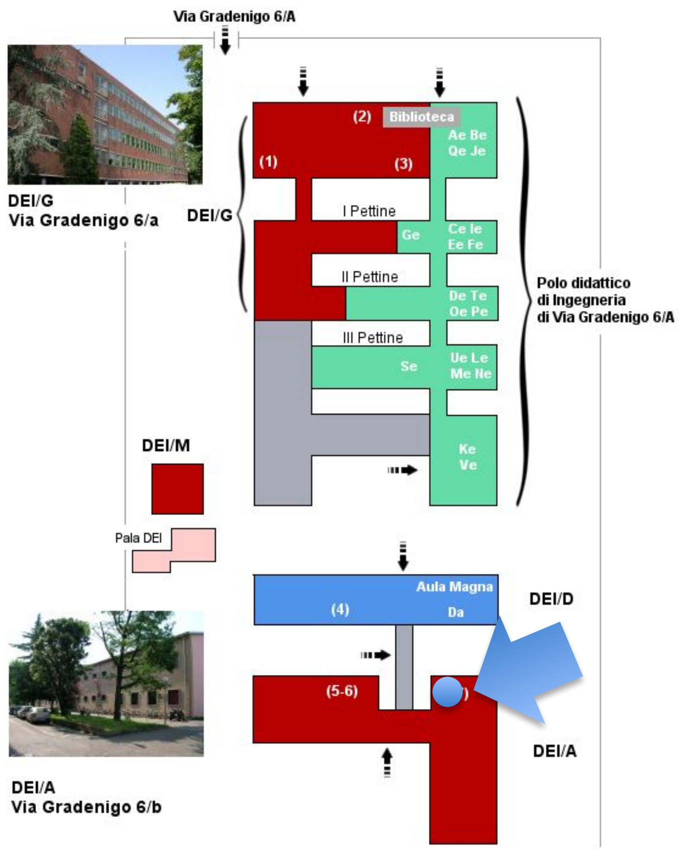
1. Programming exam (10 points)
   a. Implementation of a learning problem in Python
   b. May include algorithms from the course or general techniques (e.g., regularization or cross-validation)
   c. Open book, duration: 2 hours
2. Written exam (22 points)
   a. Four questions (including theoretical questions and exercises)
   b. Closed book, duration: 90 minutes

## Exam rules

1. In order to pass, you must get **at least 50%** on both parts
2. Partial results are valid until the end of the academic year – or until the next attempt, **even if you withdraw**
3. There will be **no oral integrations**
4. You can retake each part separately if you do not pass

## How to reach me

- ➔ Questions during the break or after class
- ➔ Email: federico.chiariotti@unipd.it (I try to reply within a day)
- ➔ If you want to set up a meeting, send me an email. My office is in room 231, DEI/A (via Gradenigo, 6/B)



## Setting up your computer

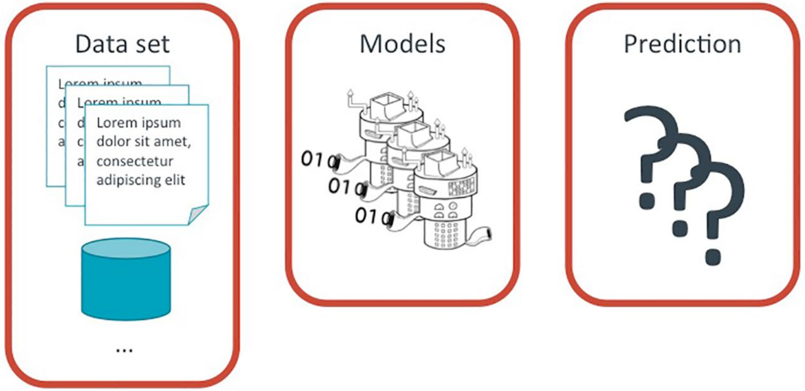
- ➔ We will use **Jupyter notebooks** as a main tool
- ➔ Recommended: install anaconda3 (https://www.anaconda.com/download)
- ➔ Jupyter notebooks are included by default in Anaconda
- ➔ Useful packages: numpy, pandas, scipy, scikit-learn, matplotlib

ANACONDA®

## Running notebooks

- ➔ Command: `jupyter notebook`
- ➔ The notebook is an interactive environment in your browser
- ➔ If you’ve never used python, it’s very intuitive and simple
- ➔ The “lab 0” experience will help you get familiar with the syntax

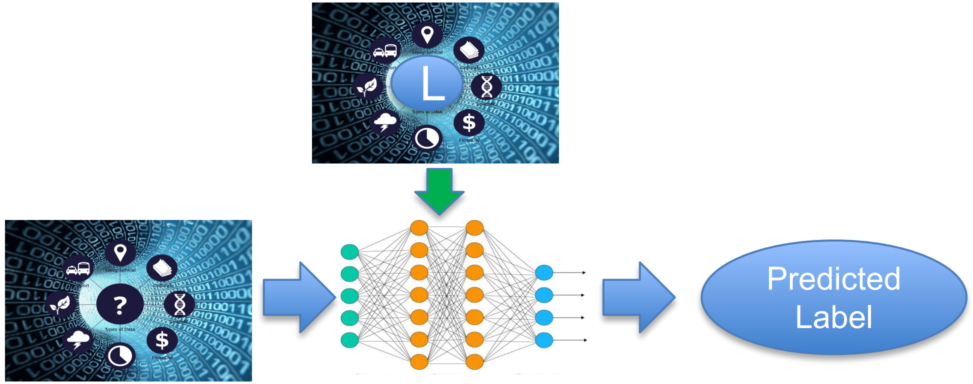
## What is machine learning?



Machine learning (ML) is a set of methods that give computer systems the ability to "*learn*" from (*training*) data to make predictions about novel data samples, *without being explicitly programmed*

# The three families of machine learning problems

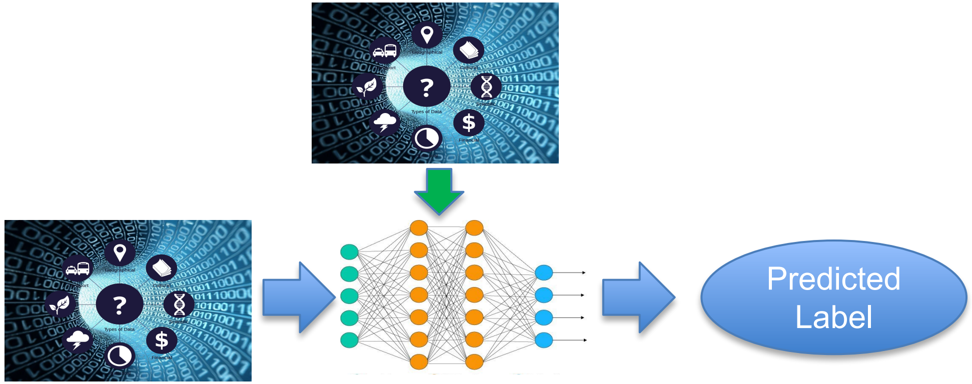
1. **Supervised learning**



➔ Classification
➔ Regression
➔ Object recognition

## The three families of machine learning problems

1. Supervised learning
2. **Unsupervised learning**

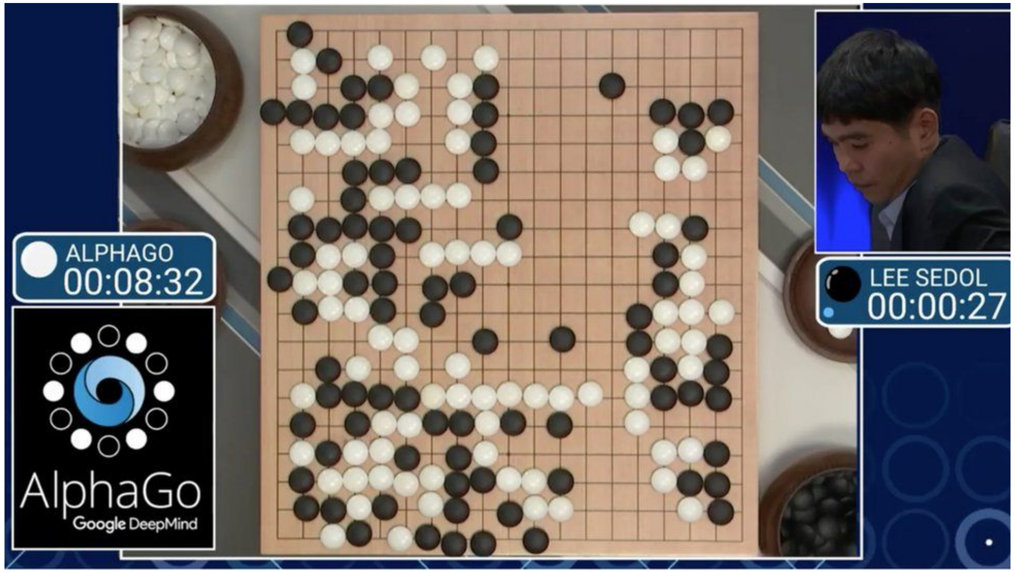


→ Clustering
→ Compression
→ Generative models

## The three families of machine learning problems

1. Supervised learning
2. Unsupervised learning
3. **Reinforcement learning**

➔ Games
➔ Robotics
➔ Sequential decision-making



## Lecture plan

| Date | # | Topic | Date | # | Topic | Date | # | Topic |
|---|---|---|---|---|---|---|---|---|
| Sep. 30 | 1 | Introduction | Oct. 31 | 7 | Linear regression | Nov. 28 | 12 | CNNs |
| Oct. 3 | 2 | PAC learning | Nov. 4 | L3 | *Linear models* | Dec. 5 | 13 | PCA |
| Oct. 7 | 3 | Uniform convergence | Nov. 7 | 8 | SVMs | Dec. 9 | L6 | *Neural networks* |
| Oct. 10 | 4 | VC dimension | Nov. 11 | 9 | Kernels | Dec. 12 | 14 | Clustering models |
| Oct. 14 | L1 | *Python basics* | Nov. 14 | 10 | Ensemble models | Dec. 9 | L7 | *Clustering* |
| Oct. 17 | 5 | Model selection | Nov. 18 | L4 | *SVMs* | Dec. 19 | 15 | Reinforcement |
| Oct. 21 | L2 | *Model selection* | Nov. 21 | 11 | Neural networks | Jan. 7 | L8 | *Reinforcement* |
| Oct. 24 | 6 | Linear classification | Nov. 25 | L5 | *Random forests* | Jan. 9 | 16 | Exercises and Q&A |

## 1-5: statistical learning

1. Basic formulation: how do we model supervised learning problems?
2. PAC learning: giving performance guarantees on specific problems
3. Uniform convergence: when can we achieve PAC learning?
4. VC-dimension: how do we quantify the complexity of a problem?
5. Model selection: how do we choose the best learning model?

## 6-12: supervised learning

6. Linear binary classification
7. Linear regression
8. SVMs: optimizing the classification margin
9. Kernel methods: tackling non-linear problems
10. Boosting and random forests: combining weak learners
11. Neural network models and gradient descent
12. Backpropagation and CNNs

## 13-15: unsupervised and reinforcement learning

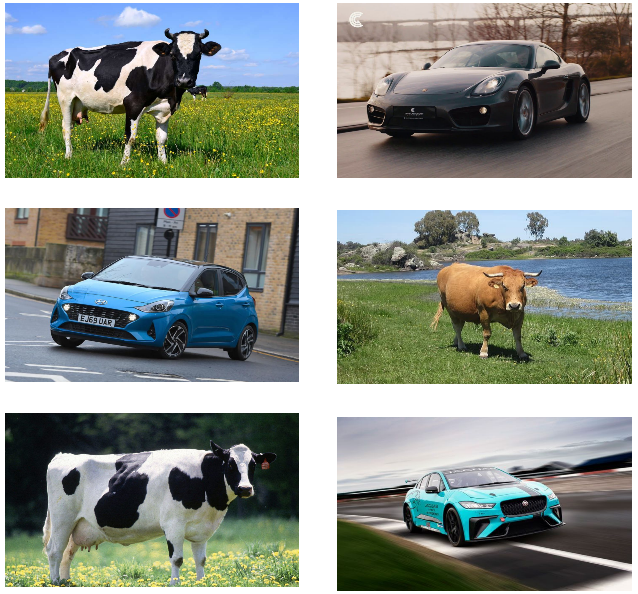
13. PCA: reducing the dimensions of data through linear algebra
14. Clustering models and algorithms: grouping objects without labels
15. Reinforcement learning: learning to act by trial and error

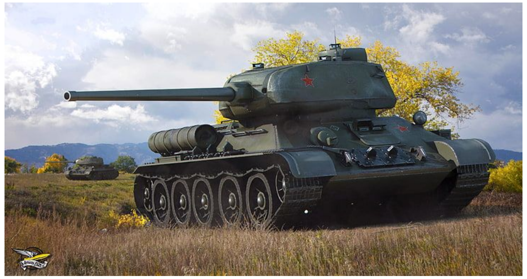
## A general idea: when not to learn

In this course, we will also see the limitations of learning approaches. There are some considerations we need to make before we start importing learning libraries and firing up python:

- ➔ Performance gains: is it worth it?
- ➔ Computational issues: learning (and deep learning in particular) is expensive
- ➔ Reliability: how can we ensure that the model will perform well in real life?
- ➔ Fairness: are we baking prejudices into our models?
- ➔ Explainability: can we understand the decisions the model makes?

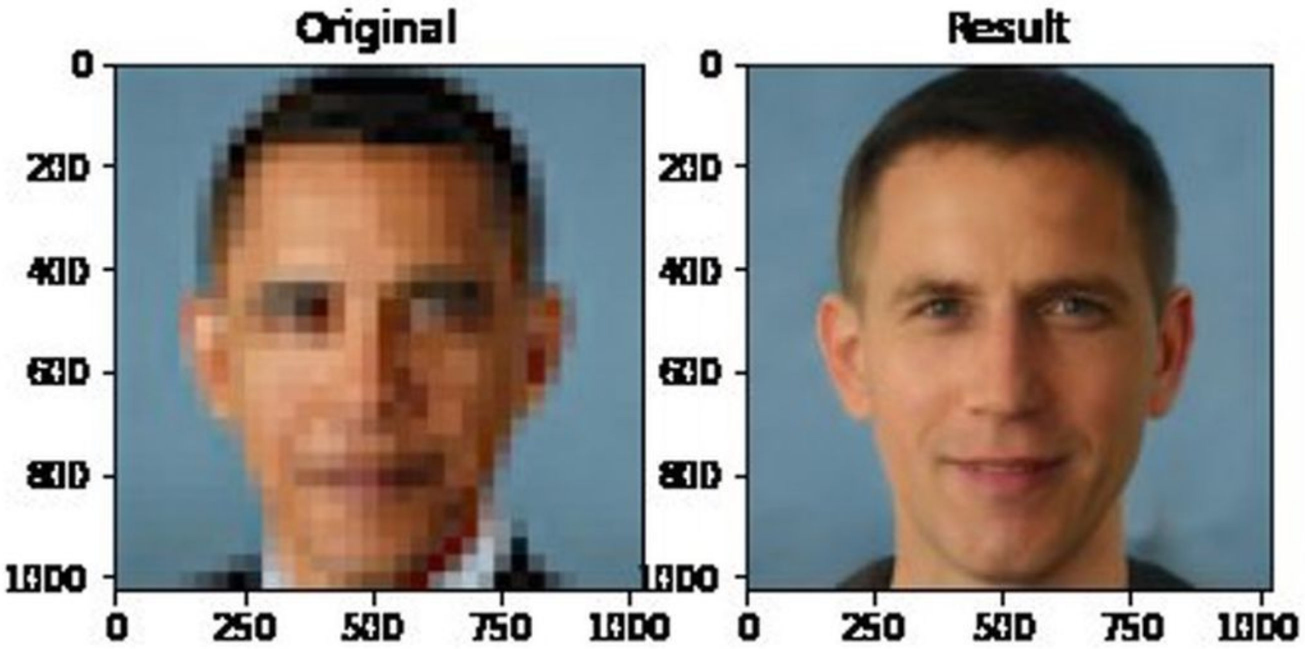
## Biases





Is this a cow?

## Biases



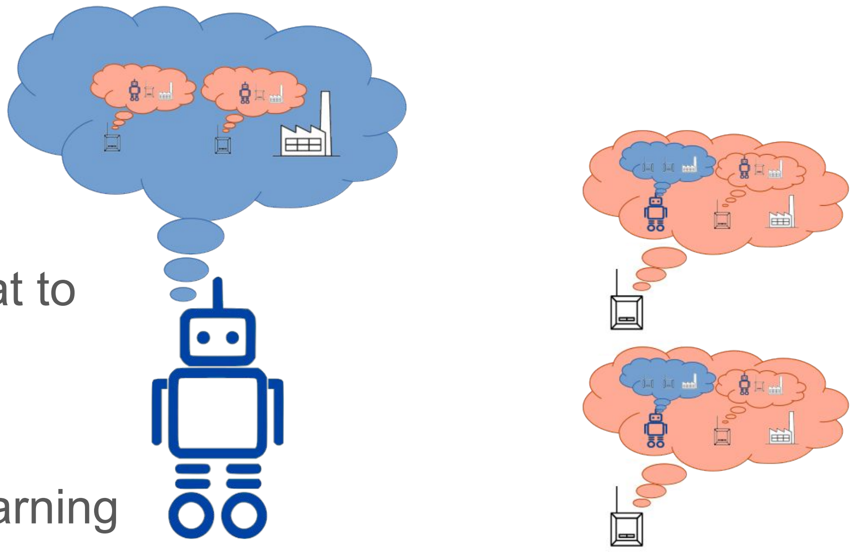
## Machine learning courses (optional, offered at DEI)

- ➔ Reinforcement Learning (1st semester)
- ➔ Adversarial Machine Learning (1st semester)
- ➔ Human Data Analytics (1st semester)
- ➔ Neural Networks and Deep Learning (2nd semester)
- ➔ Natural Language Processing (2nd semester)

## My own research

My main line of research is on **effective communication** is a new way to decide what to transmit to help an agent perform a task

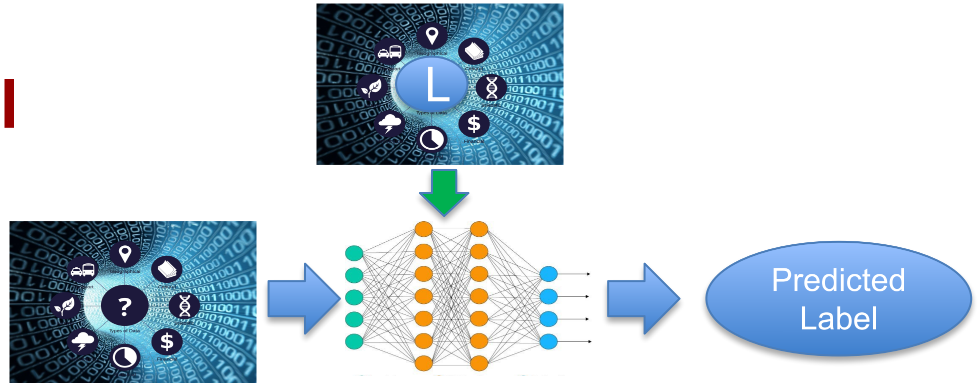
It involves a combination of unsupervised learning (to efficiently encode information) and reinforcement learning (to make decisions), combined with Markov theory, game theory, and sometimes even formal logic



## Part 3:
## Basic model formulation

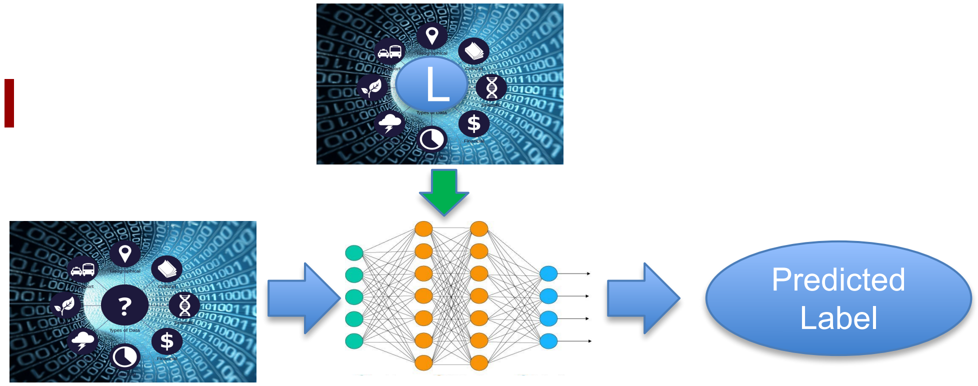
Università
degli Studi
di Padova

## Supervised learning model



- Domain set (or instance space) $X$
  - $x \in X$ is a domain point or instance
  - Usually, $x$ is represented by a tensor of *features*
- Label set $Y$
  - Simplest case: binary classification, $Y = \{0, 1\}$
- Training set $S$
  - Finite sequence of labeled points $((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$
  - Each point $x_i$ is associated to its label $y_i$

# Supervised learning model



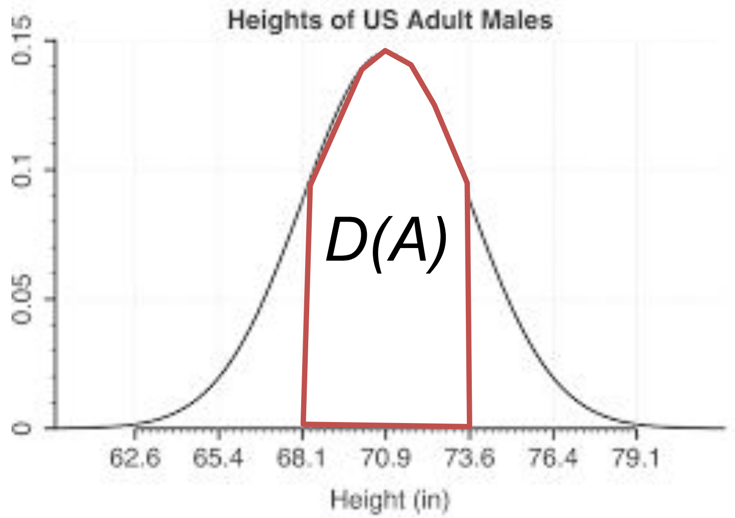
- Prediction rule $h : X \rightarrow Y$
  - The learner's output (hypothesis)
  - $A(S)$: hypothesis produced by algorithm $A$ when it is fed training set $S$
- Data generation model
  - $D$ is a distribution over $X$ **(unknown to the machine learning algorithm)**
  - Instances are labeled according to $f : X \rightarrow Y$ **(unknown to the ML algorithm)**
  - Training set: sampling according to $D, y_i = f(x_i) \ \forall x_i \in S$
- Success metric
  - Classifier error: probability of predicting the wrong label over $D$

## Events and data generating distributions

Samples $x \in X$ are produced according to a Gaussian distribution $D$

Event $A : x \in [68.1, 73.6]$

$D(A)$ is the probability of observing a point in subset $A$



$$\pi(x) = \begin{cases} 1, & \text{if } 68.1 < x < 73.6; \\ 0, & \text{otherwise.} \end{cases}$$

$$P_{x \sim D}[\pi(x) = 1] = D(A)$$

## Loss functions

- ➔ Assume a domain subset $A \subset X$
- ➔ $A$ is an event expressed by $\pi : X \to \{0,1\}$, i.e., $A = \{x \in X : \pi(x) = 1\}$
- ➔ We get $P_{x\sim D}[\pi(x) = 1] = D(A)$

Error of prediction rule $h : X \to Y$

$$L_{D,f}(h) \overset{\text{def}}{=} P_{x\sim D}[h(x) \neq f(x)] = D(x : h(x) \neq f(x))$$

$L_{D,f}(h)$ (often written just as $L_D(h)$) is called true loss, true risk, or generalization loss

## Empirical Risk Minimization (ERM)

- ➔ Learner outputs $h_S : X \to Y$
- ➔ **Goal: find $\mathbf{h}^*$ that minimizes $\mathbf{L_{D,f}(h)}$**
- ➔ Both $D$ and $f$ are unknown!

ERM: we minimize the loss on the training set $L_S(h) = \frac{\sum_{i=1}^{m} |h(x_i) - y_i|}{m}$

This works if we use a 0-1 loss: the definition can be generalized

The empirical risk is also called training error or training loss

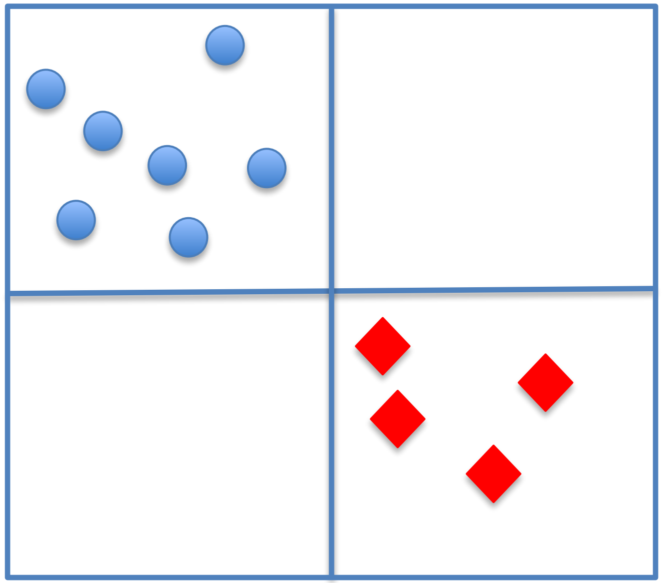
## Can we trust training error?

Let us consider the training set S in the figure, with red squares indicating class 1

$h_s(x_1,x_2)=I(x_1>0)$

This function is 1 if the point is on the right side.

$L_S(h_s)=0$

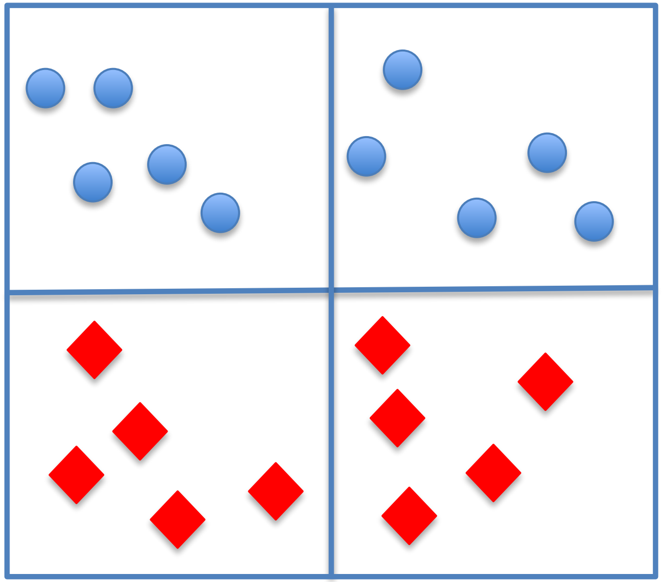


## Can we trust training error?

The real data distribution is different: X is drawn uniformly, but the rule is orthogonal

$f(x_1,x_2)=I(x_2<0)$

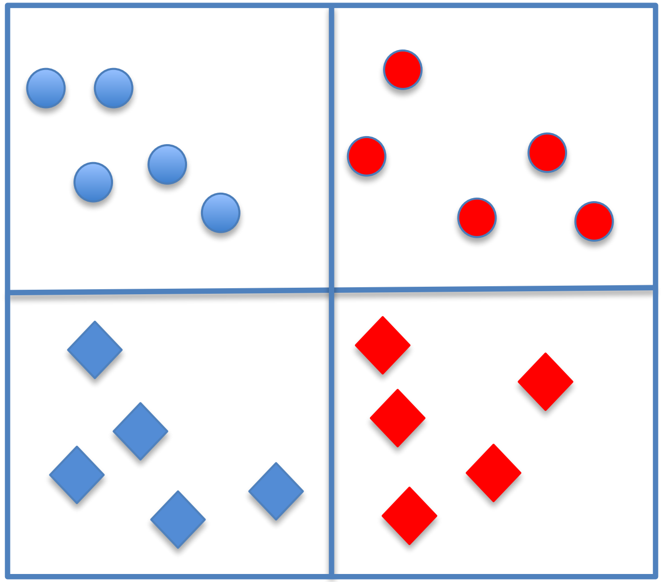
This function is 1 if the point is on the lower side.



## Can we trust training error?

In this case, our loss is exactly ½, the same as a completely random guess!

When does ERM lead to good generalization performance (i.e., a low true error?)



## Hypothesis classes

We can apply ERM over a *restricted* class $H$ of possible hypotheses

$h \in H$ is a function $h : X \to Y$

Restricted ERM finds $h^* \boxed{\in} \arg\min_{h \in H} L_S(h)$

There might be multiple solutions

How can we choose a class that avoids overfitting?

## Assumptions

1. Let us start with a *finite* class $H$
2. Let $h_S = \mathrm{ERM}_H \in \arg\min_{h \in H} L_S(h)$

We need two further assumptions:

3. We are lucky, and have *realizability*: $f \in H$, i.e., $\exists h^* \in H : L_{D,f}(h^*) = 0$
4. IID assumption: samples are drawn independently from $D$

Can we learn $h^*$ with ERM?