

# Deep Neural Networks

## Lecture 1

# Format

- Lecture + Labs (hands-on)
- Instructors: Robert Bogucki, Marek Cygan, Maciej Jaśkowski, Maciek Klimek, Marcin Mucha, Marcin Pilipczuk
- Slack: [dl1718.slack.com](https://dl1718.slack.com)
- Slides will be available (<http://bit.ly/2opx3kg>)

# Grading

- 65% hands-on assignments
- 35% oral exam
- You need to **pass both**

Two options for hands-on assignments:

- 3 small projects,
- 1 big project (teaming up is allowed, max 3 people per team)

# Hands-on assignments: big project

- Apply Deep Learning to an interesting problem
- Required deliverables:
  - Presentation during the last lecture of the course
  - Short writeup
  - Code
- Be sure to have your project proposal accepted before you decide to pursue it
- You can work in teams of up to 3 people
- Projects with more people are expected to deliver more impressive results

# Hands-on assignments: big project

- Check those for some inspirations:  
<http://cs231n.stanford.edu/project.html>  
<http://cs229.stanford.edu/projects2013.html>  
<http://cs231n.stanford.edu/reports.html>
- Kaggle may be a good idea as well:  
e.g. <https://www.kaggle.com/>

# Books, tutorials, ...

- <http://www.deeplearningbook.org/>
- Convolutional Neural Networks for Visual Recognition  
<http://cs231n.stanford.edu/>
- [Kaggle.com](https://www.kaggle.com/)
- Machine Learning: A Probabilistic Perspective (Kevin Murphy)

# A short survey

- What do you know about ML?
- Why do we need ML?

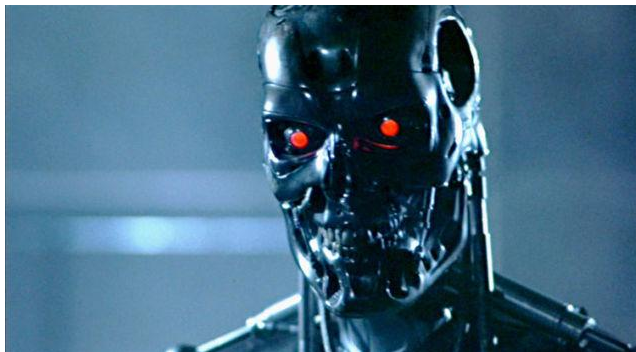
# A short survey

- What do you know about ML?
- Why do we need ML?
- DL?

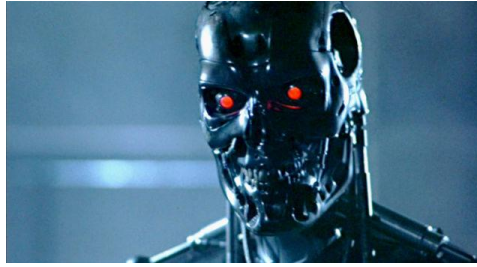
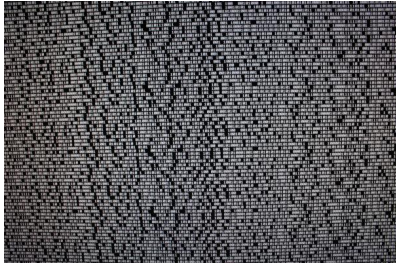


# Plan for the next two weeks

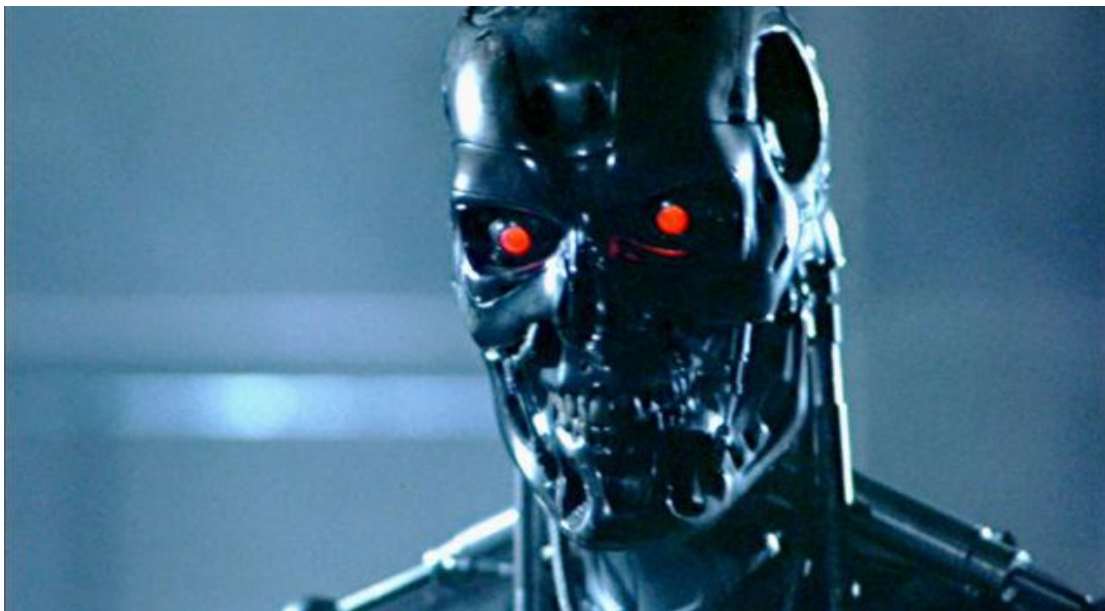
- A refresher on Machine Learning fundamentals



# How computers learn to do things?



# AI demystified



$$= [w_1, w_2, \dots, w_n]$$

# A simple example

$$\textit{apartment's price} \approx w_1 \cdot \textit{area} + w_2 \cdot \textit{district} + \dots$$

# A simple example

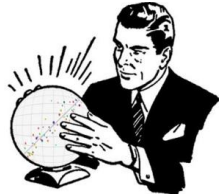
$$\text{apartment's price} \approx w_1 \cdot \text{area} + w_2 \cdot \text{district} + \dots$$

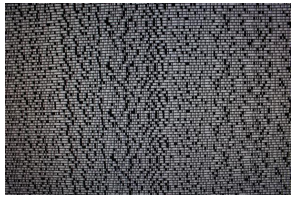


If you know those, you can predict the price!



$[area, district, ...]$





$[area, district, ...]$



$[w_1, w_2, ..., w_n]$





$[area, district, ...]$



$[w_1, w_2, ..., w_n]$



$apartment's\ price \approx w_1 \cdot area + w_2 \cdot district + ...$



# Man vs Machine

*apartment's price  $\approx w_1 \cdot \text{area} + w_2 \cdot \text{district} + \dots$*

*apartment's price  $\approx w_1 \cdot \text{area} \cdot \text{average price in the district per sq. meter} + \dots$*

# Man vs Machine

*apartment's price  $\approx w_1 \cdot \text{area} + w_2 \cdot \text{district} + \dots$*

*apartment's price  $\approx w_1 \cdot \text{area} \cdot \text{average price in the district per sq. meter} + \dots$*


- Which **model** should we choose?
- How to come up with a **model**?
- Area, district, average price, ...?
- What should **w1**, **w2** be equal to?


# Man vs Machine

*apartment's price  $\approx w_1 \cdot \text{area} + w_2 \cdot \text{district} + \dots$*

*apartment's price  $\approx w_1 \cdot \text{area} \cdot \text{average price in the district per sq. meter} + \dots$*

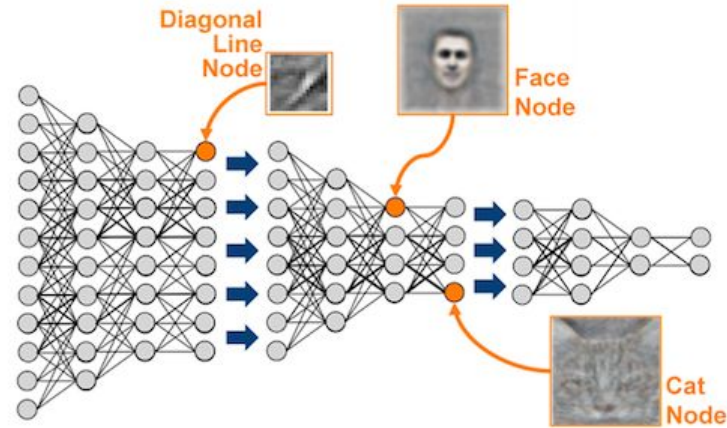
- Which **model** should we choose?
- How to come up with a **model**?
- Area, district, average price, ...?
- What should **w1**, **w2** be equal to?

 (typically) a **data scientist's** job

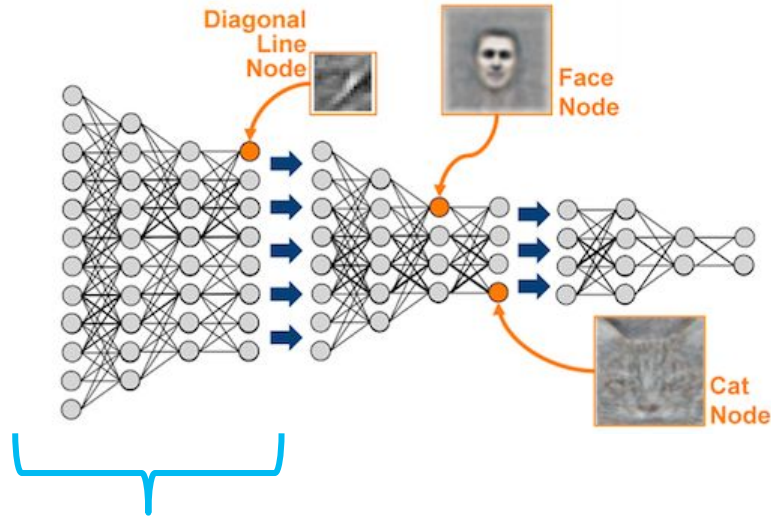
 comes with the historical data

- (typically) a **machine's** job  
(based on the historical data)

# Why should we care about Deep Learning?



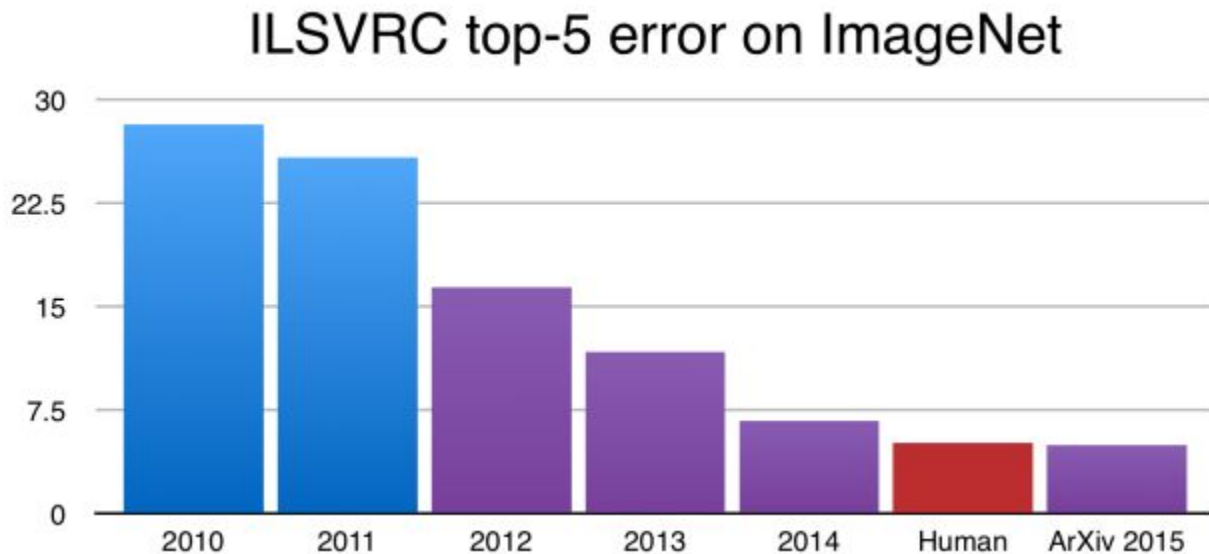
# Why should we care about Deep Learning?



humans did this part manually just a few years ago (2011)...

# Why should we care about Deep Learning?

...and we're not that good at it.



# Classification



VS



# Classification



VS

ELIE WIESEL  
555 MADISON AVENUE, 20TH FLOOR  
NEW YORK, NY 10022

His Excellency  
László Kövér  
Speaker of the Hungarian National Assembly  
Hungarian National Assembly  
Kossuth tér 1-3  
H-1055 Budapest  
Hungary

June 7, 2012

Mr. Speaker:

It is with profound dismay and indignation that I learned of your participation, together with Hungarian Secretary of State for Culture Géza Szűcs and far-right Jobbik party leader *Gábor Vona*, in a ceremony in Romania honoring *József Nyírő*, a member of the National Socialist Arrow Cross Parliament. I found it outrageous that the Speaker of the Hungarian National Assembly could participate in a ceremony honoring a Hungarian fascist ideologue of the Horthy and *Szalasi* regimes. This distressing news came following the resurgent practice of naming public spaces after wartime leader Miklós Horthy and of rehabilitating Albert Wass and other figures that collaborated heavily with the Hungarian fascist regime. I was also informed that the writings of extreme right intellectuals are systematically introduced in the Hungarian curriculum.

In a session held in the Hungarian Parliament on December 9, 2009, I urged your colleagues "to do even more to denounce antisemitic elements and racist expressions in your political environment and in certain publications... I believe that they bring shame to your nation..."

Since that time it has become increasingly clear that Hungarian authorities are encouraging the whitewashing of tragic and criminal episodes in Hungary's past, namely the wartime Hungarian governments' involvement in the deportation and murder of hundreds of thousands of its Jewish citizens.

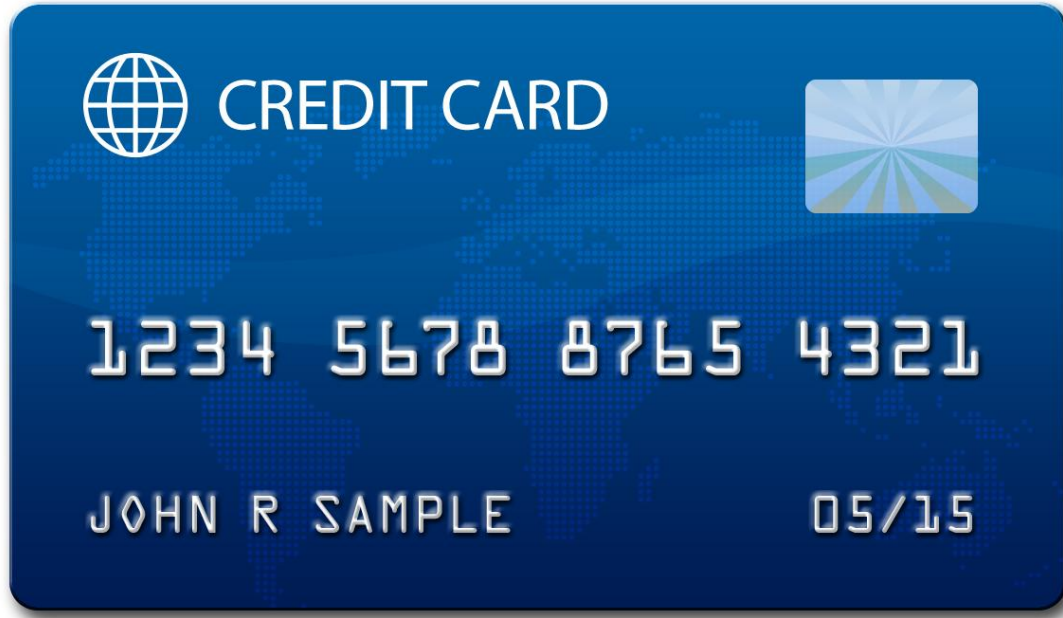
I do not wish to be associated in any way with such activities. Therefore, I hereby repudiate the Grand Cross Order of Merit of the Republic of Hungary granted to me on June 24, 2004, by the President of Hungary.

Sincerely,

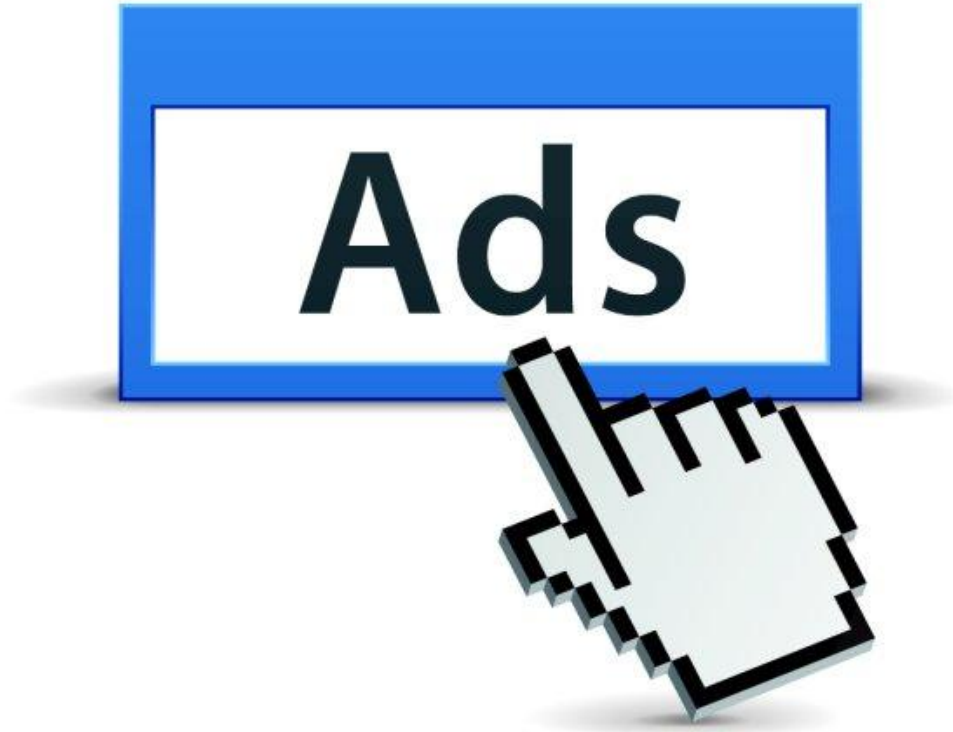
  
Elie Wiesel



# Classification



# Classification



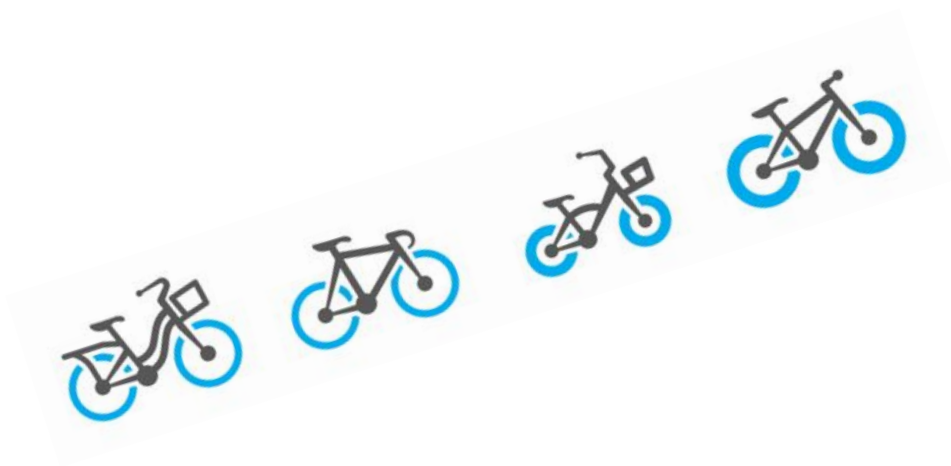
# Classification

4	0	1	0	5	0	7	3
5	2	7	0	5	4	2	2
3	5	7	2	6	4	5	4
3	5	4	2	4	7	4	5
5	5	3	0	8	8	2	2
0	4	0	3	1	5	9	8
4	0	6	9	7	7	4	3
6	6	9	1	3	4	8	7

# Classification



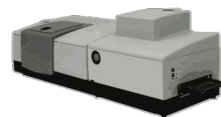
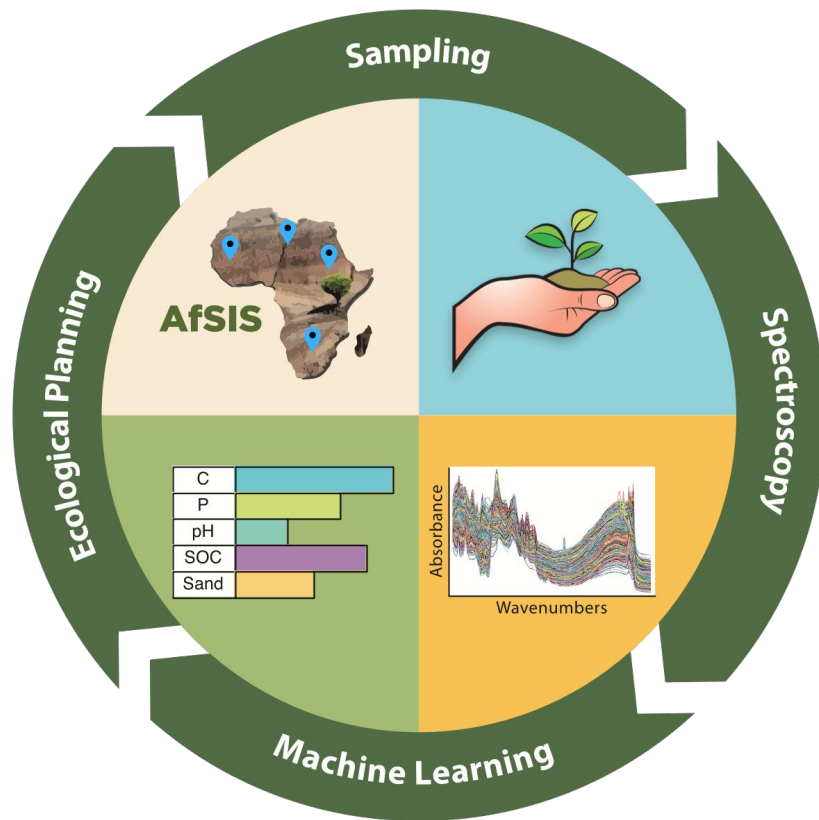
# Regression



# Regression



# Regression



Which one is it?





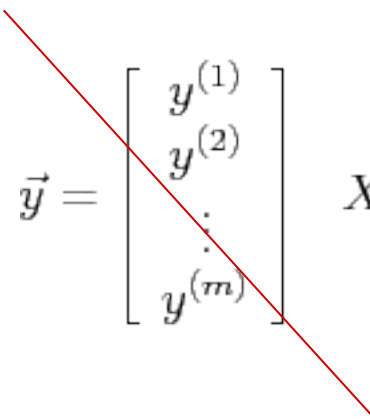
# Supervised learning

Classification, regression

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad X = \begin{bmatrix} \text{---}(x^{(1)})^T\text{---} \\ \text{---}(x^{(2)})^T\text{---} \\ \vdots \\ \text{---}(x^{(m)})^T\text{---} \end{bmatrix}$$

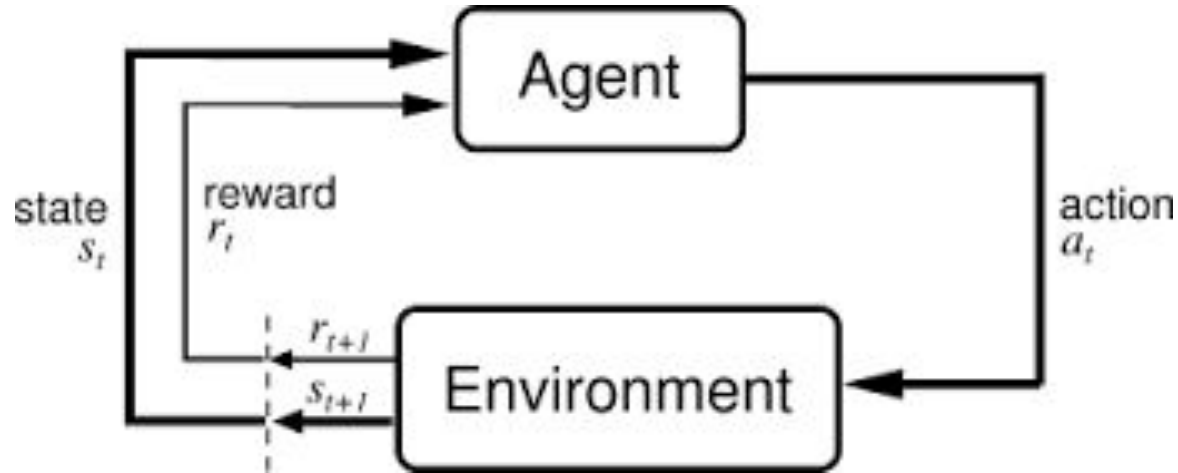
# Unsupervised learning

Clustering, dimensionality reduction


$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad X = \begin{bmatrix} -(x^{(1)})^T \\ -(x^{(2)})^T \\ \vdots \\ -(x^{(m)})^T \end{bmatrix}$$

# Reinforcement learning

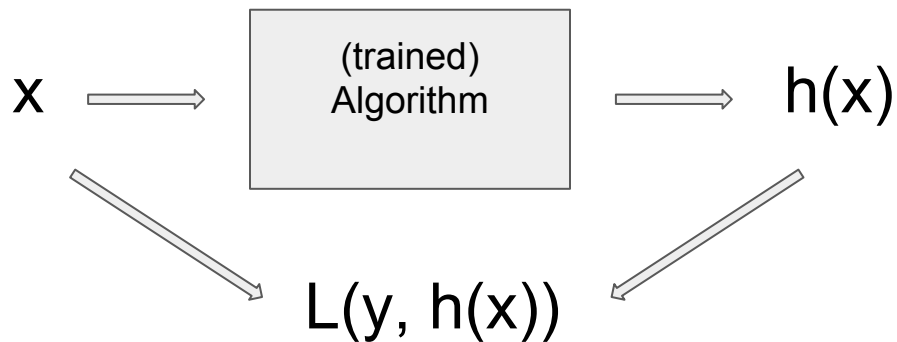
Game-like environment, sequential decision making



# How to judge our algorithm?

L - Loss function

$(x, y)$  - a single instance (desired mapping)



(we usually look at the mean:  $1/m \sum L(y_i, h(x_i))$ )

# Examples of loss functions

## Regression

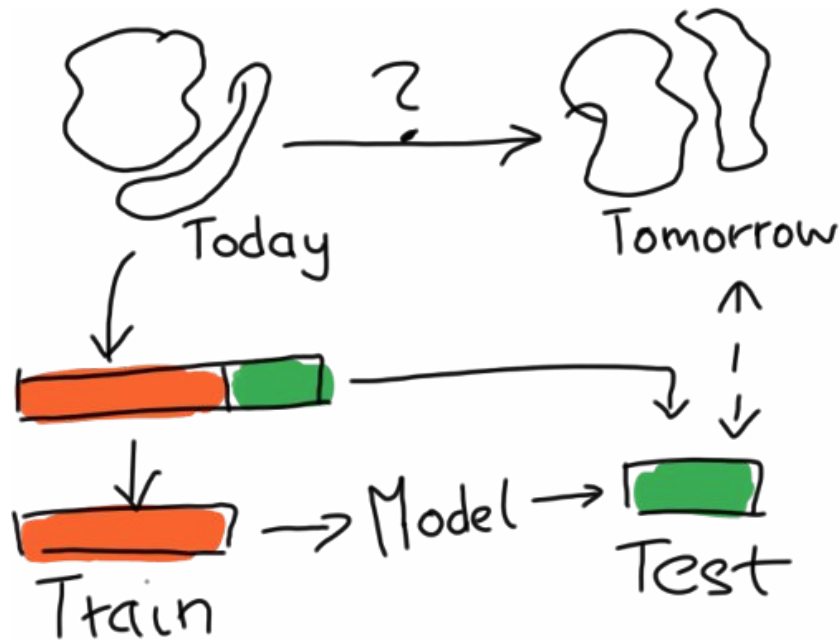
- $(y - h(x))^2$
- $|y - h(x)|$
- $(|y - h(x)| - \epsilon) I(|y - h(x)| > \epsilon)$

## Classification

- $I(y \neq h(x))$
- $-y \log h(x) - (1 - y) \log(1 - h(x))$
- top5 error

# Loss function is the protagonist here

We would like to minimize it “in the future”



# Machine Learning demystified

**MINIMIZE THE LOSS FUNCTION**



# Why should we care about the loss function?

Let  $h(x) = \text{const}$ , what constant should we choose?

## Regression

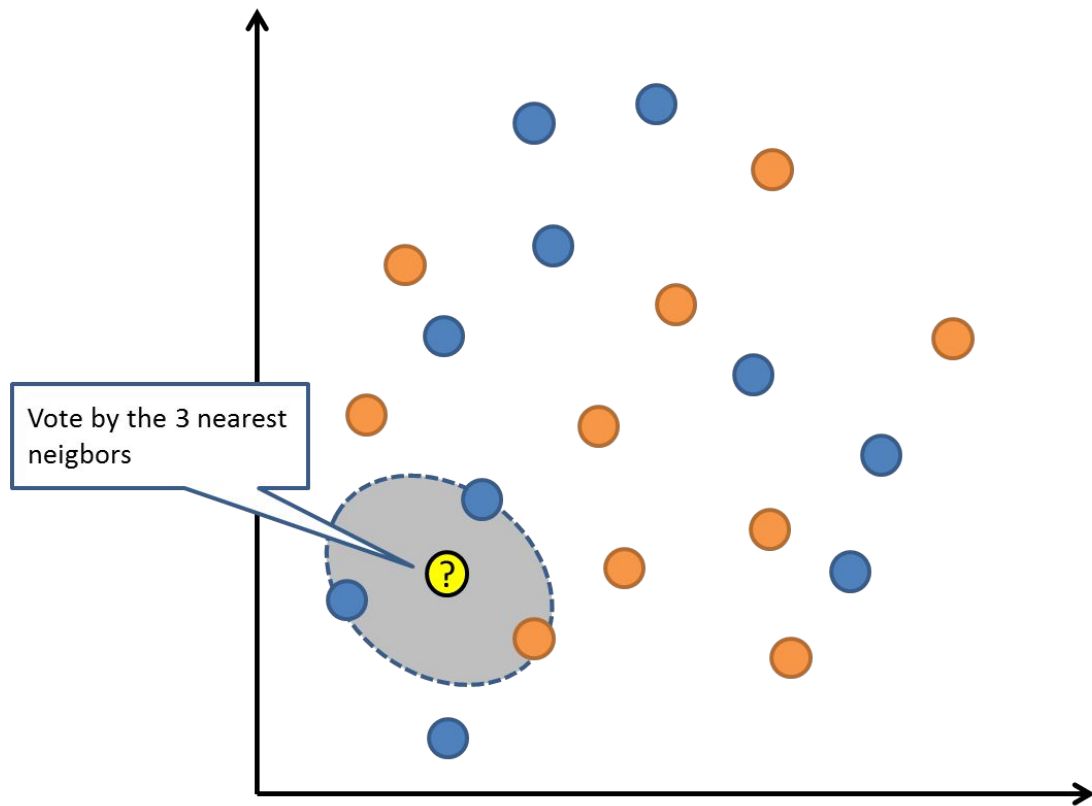
- $(y - h(x))^2$
- $|y - h(x)|$
- $(|y - h(x)| - \epsilon) \mathbb{I}(|y - h(x)| > \epsilon)$

## Classification

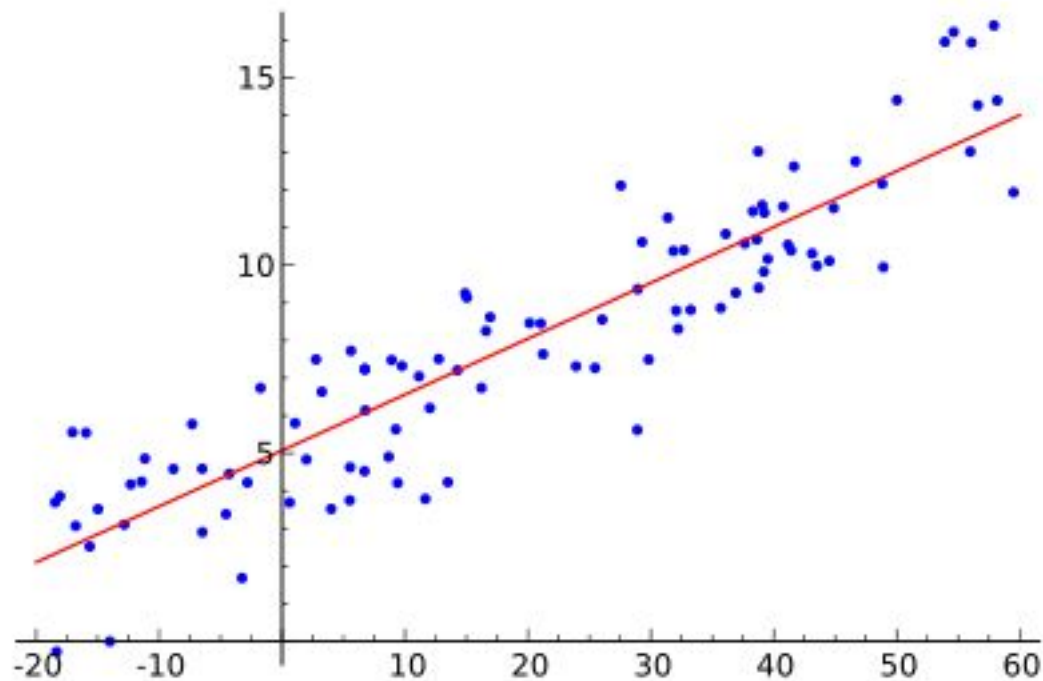
- $\mathbb{I}(y \neq h(x))$
- $-y \log h(x) - (1 - y) \log(1 - h(x))$
- top5 error



# kNN



# Linear regression



# Linear regression

- $x$  is a vector of so called features describing our instance
- $h(x) = w^T x$  (assuming  $x_0 = 1$ )
- We minimize RMSE (root mean square error):

$$\sqrt{\frac{1}{n} \sum_{k=1}^n (y_i - h(x_i))^2}$$

# Linear regression

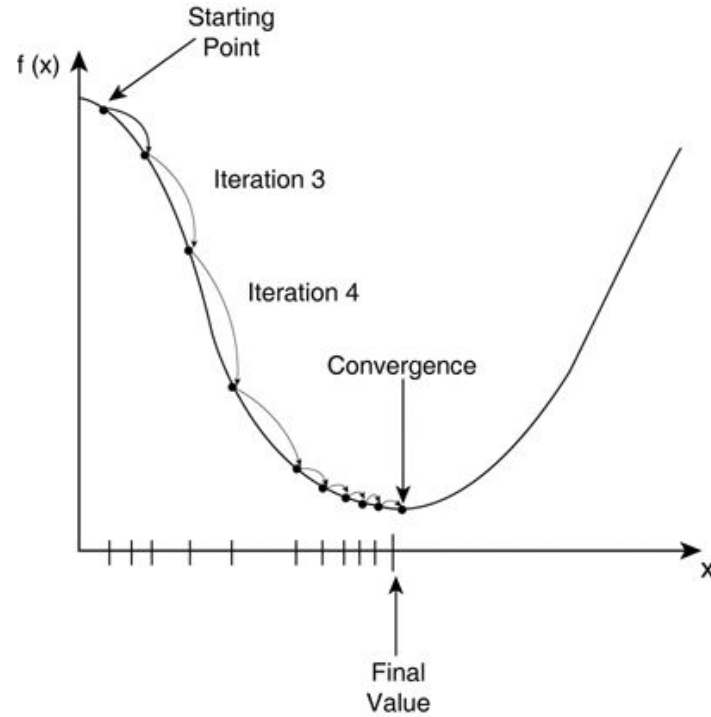
How to find  $w$ ?

# Linear regression

How to find  $w$ ?

(Naive ideas are welcome)

# Gradient descent



# Gradient descent

$$J(w) = \frac{1}{n} \sum_{k=1}^n (y_i - w^T x_i)^2$$

$$\nabla J(w) = ?$$

# Finding the right weights

Gradient descent:

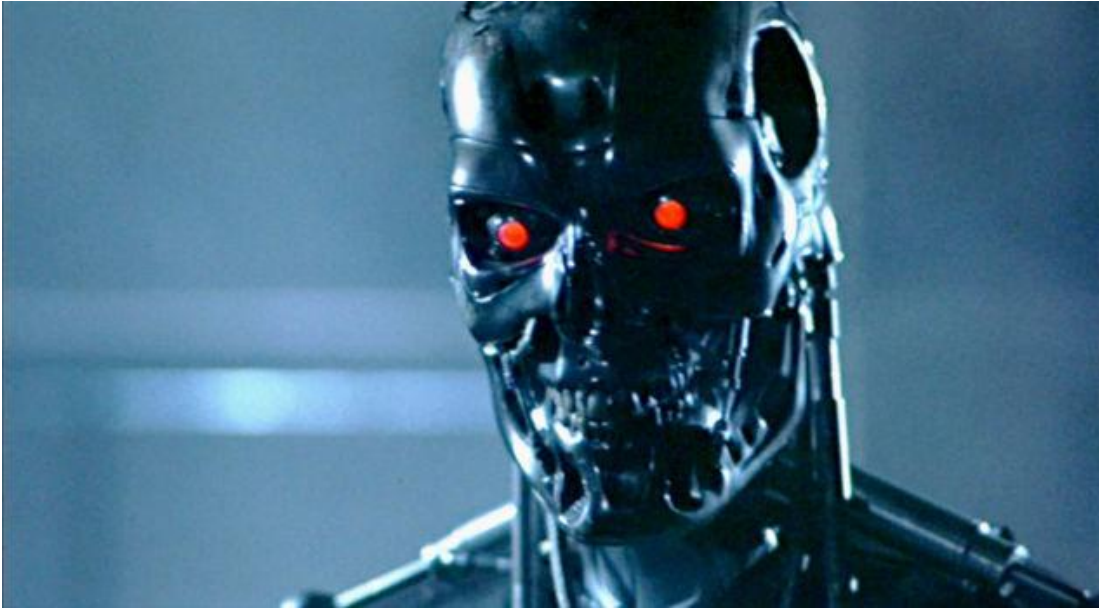
$$w_j^{(t+1)} := w_j^{(t)} - \alpha \frac{\partial J(w)}{\partial w_j}$$

Exact formula:

$$w = (X^T X)^{-1} X^T y$$

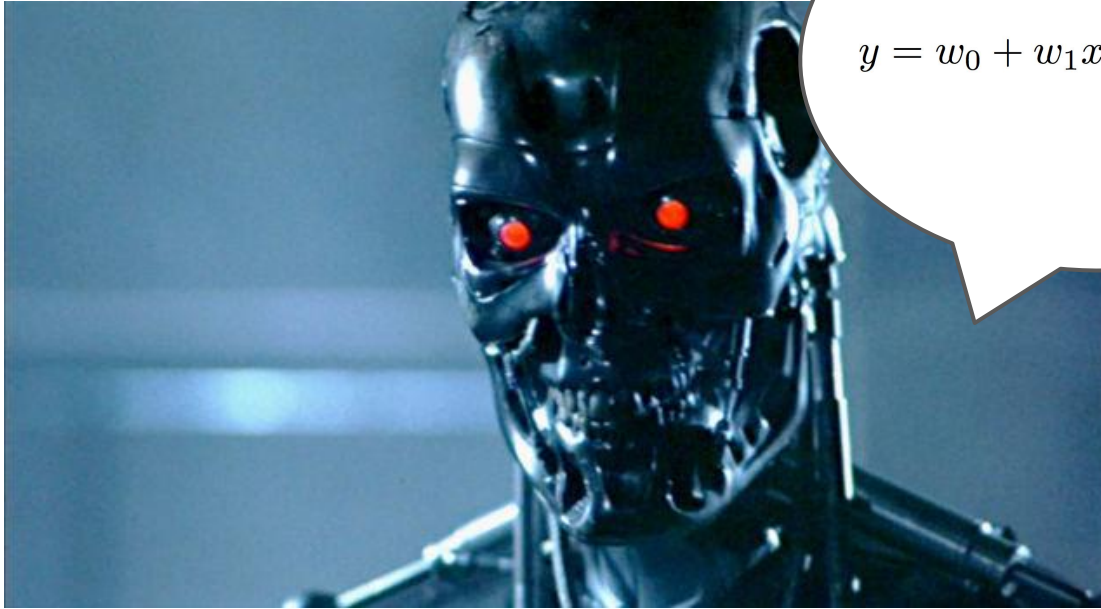


# “Intelligent” system revisited



$$= [w_0, w_1, \dots, w_n]$$

# “Intelligent” system revisited



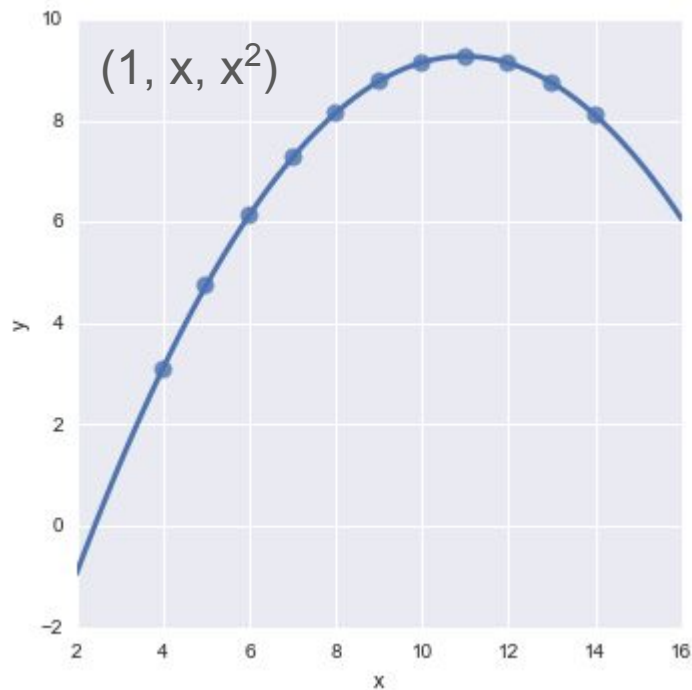
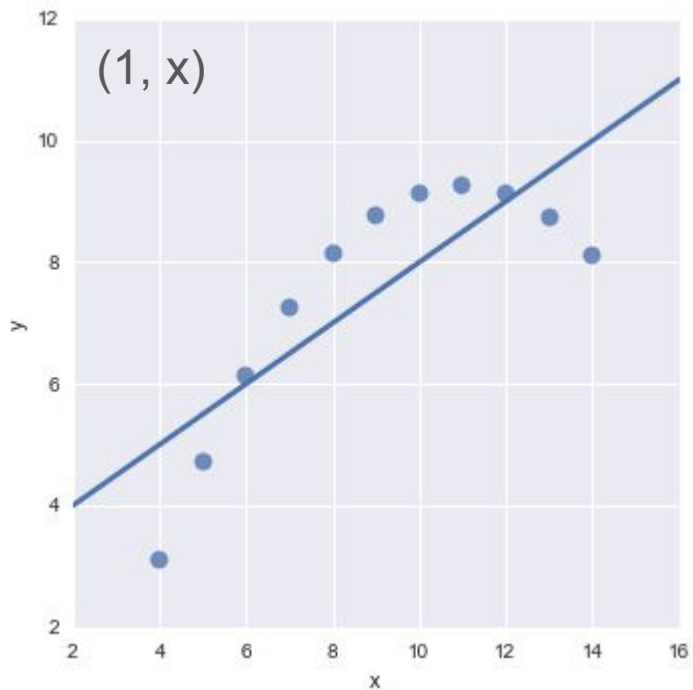
$$y = w_0 + w_1x_1 + \dots + w_nx_n$$

Beyond linearity - feature engineering!

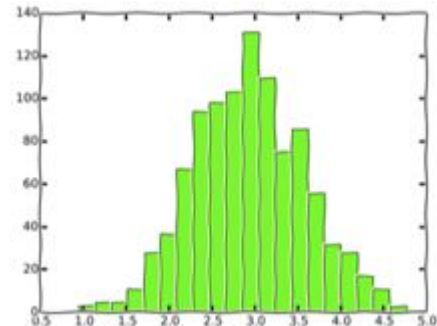
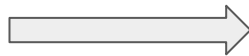
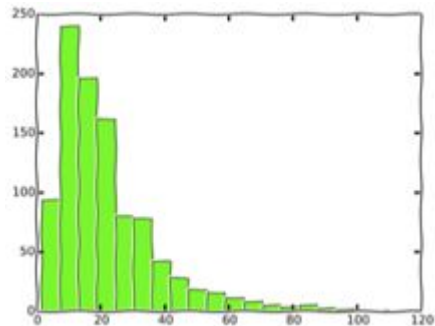
$$x \longrightarrow \phi(x)$$

$$w^T x \longrightarrow w^T \phi(x)$$

# The devil is in the features



# The devil is in the features



The devil is in the features

id	...	ulubione zwierzę	...
1	...	wombat	...
2	...	jenot	...
3	...	pies	...
4	...	kot	...
...	...	...	...

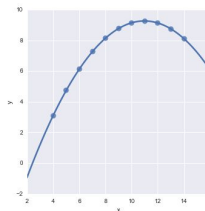
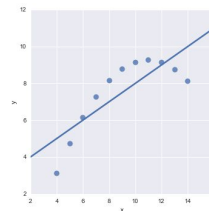
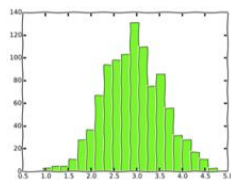
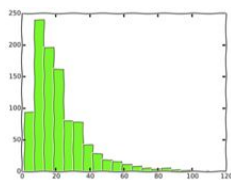
# Feature engineering

This is often the most tedious but the most rewarding (score-wise) part!

$$x \rightarrow \phi(x)$$

$$w^T x \rightarrow w^T \phi(x)$$

id	...	ulubione zwierzę	...
1	...	wombat	...
2	...	jenot	...
3	...	pies	...
4	...	kot	...
...	...	...	...



# Recap

$x_1, x_2, \dots, x_n \in R^D$  (features)

$y_1, y_2, \dots, y_n \in R$  (targets)

Goal: find  $w_0, w_1, \dots, w_D \in R$  so that our prediction

$$h(x) = w^T x$$

is good. For now, our proxy for goodness is the MSE loss function:

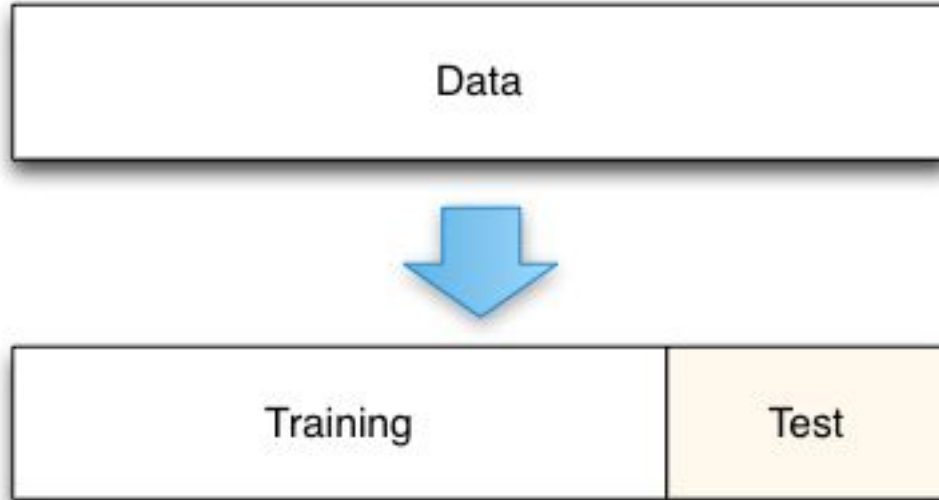
$$J(w) = \frac{1}{n} \sum_{i=1}^n (y_i - w^T x_i)^2.$$



# Some questions...

- How do we choose the right algorithm?
- How do we choose the right hyperparameters?
- How do we anticipate the future performance?

# Train/Test split

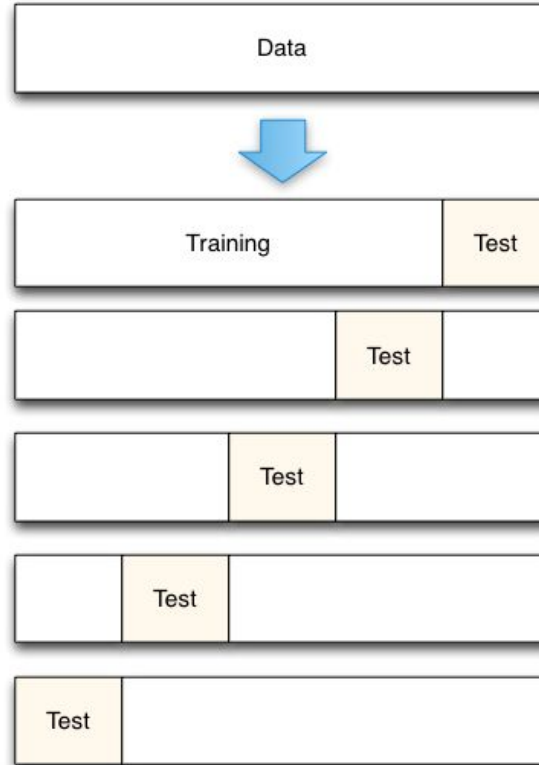


# Train/Validation/Test split

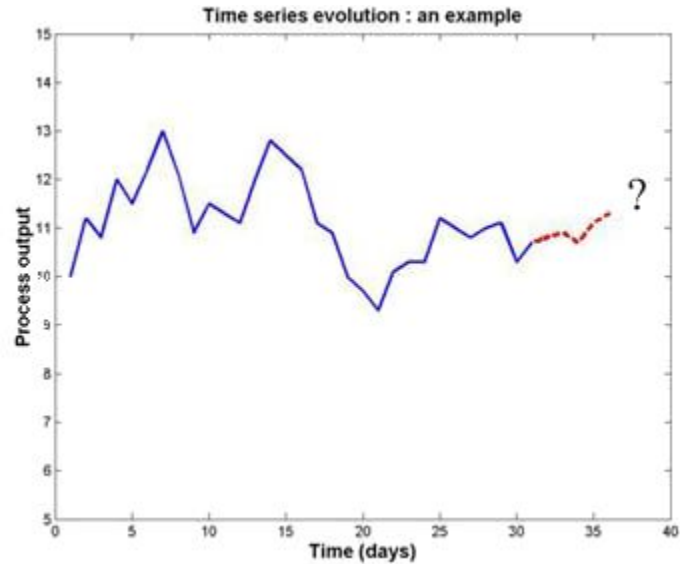
Useful if you need to anticipate the score on top of optimizing the hyperparameters

Train	Validate	Test
-------	----------	------

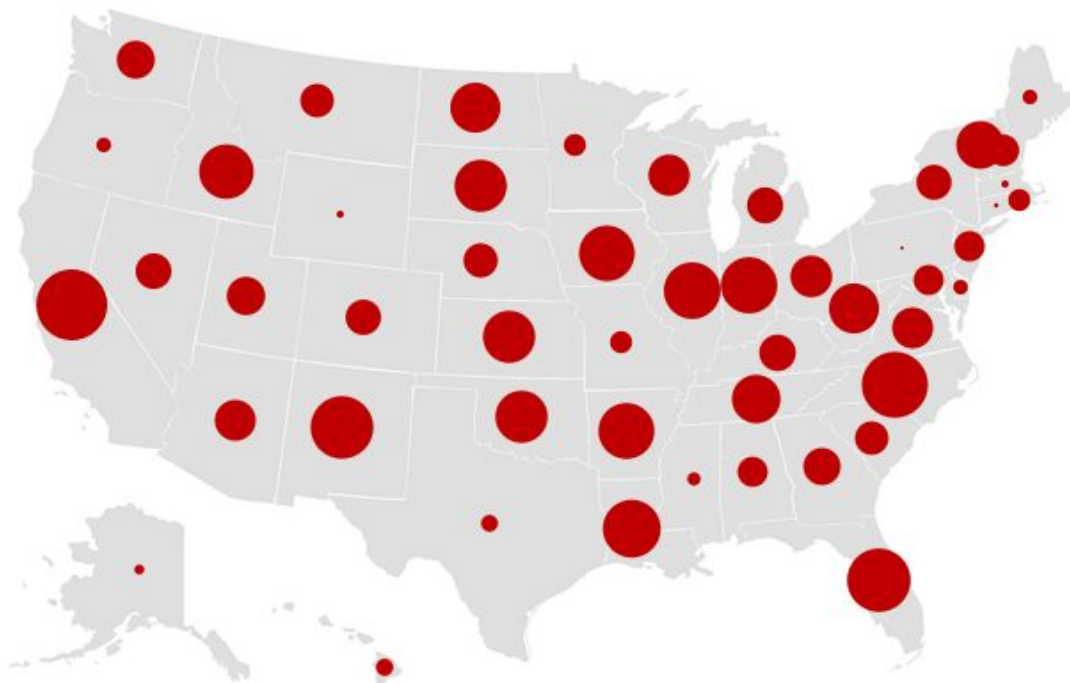
# Cross validation



# It's a Trap! #1



## It's a Trap! #2



# It's a Trap! #3

<feature selection.ipynb>

# Where to read more about it?

- Machine Learning A Probabilistic Perspective (1. Introduction)  
(recommended)  
<http://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-22may12.pdf>
- An Introduction to Statistical Learning (2. Statistical Learning)  
<http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Fourth%20Printing.pdf>
- Elements of Statistical Learning (2. Overview of Supervised Learning)  
[http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII\\_print10.pdf](http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf)



# Where to read more about it?

- Machine Learning A Probabilistic Perspective (7. Linear Regression) (math heavy + we'll cover regularization next week anyway)
- An Introduction to Statistical Learning (3. Linear Regression)  
<http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Fourth%20Printing.pdf>
- 10 things I wish I knew... ...about Machine Learning Competitions -  
[http://people.inf.ethz.ch/jaggim/meetup/slides/ML-meetup-9-vonRohr-kaggle.p](http://people.inf.ethz.ch/jaggim/meetup/slides/ML-meetup-9-vonRohr-kaggle.pdf)  
[df](http://people.inf.ethz.ch/jaggim/meetup/slides/ML-meetup-9-vonRohr-kaggle.pdf) and <http://blog.kaggle.com/2014/08/01/learning-from-the-best/> (I don't agree with everything, but might be a good start if you plan to compete on Kaggle)