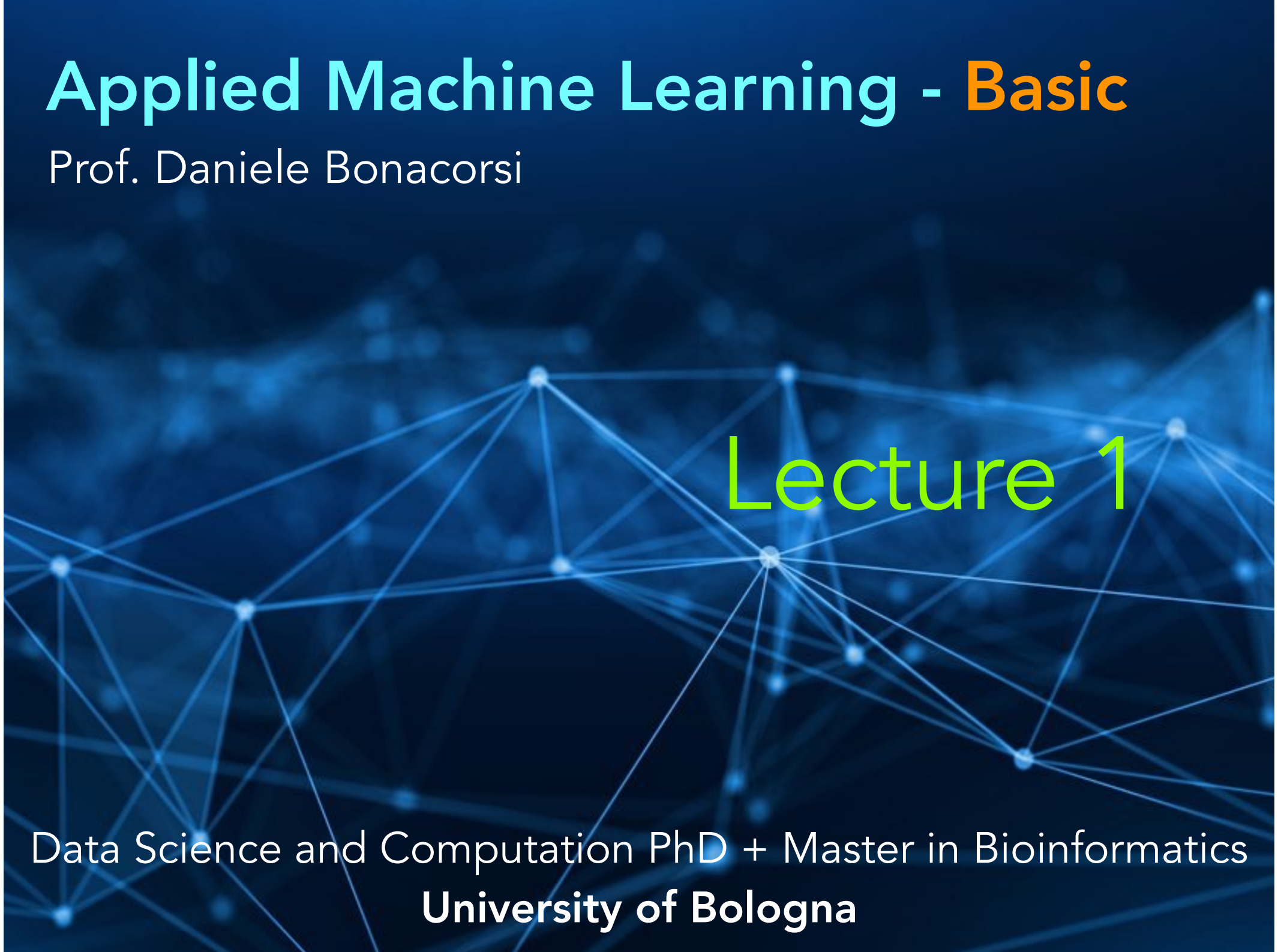
Applied Machine Learning - Basic

Prof. Daniele Bonacorsi



Lecture 1

Data Science and Computation PhD + Master in Bioinformatics

University of Bologna


Definition(s) of ML

Definition(s) of Machine Learning

"The capacity of a computer to learn from experience, i.e. to modify its processing on the basis of newly acquired information"

– The Oxford dictionary of statistics terms (today)

"ML is the field of study that gives computers the ability to learn without being explicitly programmed"

– Arthur Samuel (1959), author of the Samuel Checkers-playing Program (and some TeX..)

"A machine 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."

– Tom Mitchell (1997), [MIT1]

How many more?

Even among ML practitioners there isn't a unique, universally accepted best way to define what ML is in words

- You have read dozens yourself, perhaps. Just google for it..

But the ML community agrees on what ML is. Back on Mitchell's one, which is remarkably interesting, and thinking on a checkers/chess playing example:

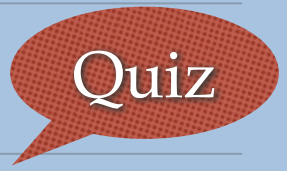
- **T** = the task of playing checkers
- **E** = the experience of playing many games of checkers (against itself)
- **P** = the probability that the program will win the next game (against somebody else)

Read it again now, in this example:

"A machine is said to learn from **(E) playing many games** with respect to **(T) the task of playing that game** and the **(P) probability to win** **if the ability to play, measured by the number of wins, improves as more and more games are played**"

Is it clear enough, at the level we can apply the definition to another example?

C.Y.U.



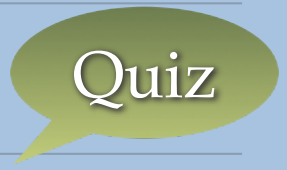
Applying Mitchell definition to another example.

Your e-mail system is monitoring which emails you do (or do not) flag as spam. Based on the collected info, it learns over time how to flag (and filter out) spam e-mails for you.

What is the task T, E, P in this example, among the following?

- classifying emails as spam or not spam
- watching you label emails as spam or not spam
- the # or fraction of emails correctly classifieds spam/not spam
- none of the above - this is not a ML problem

C.Y.U.



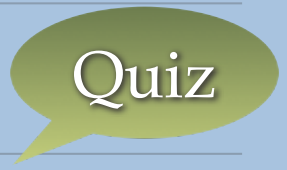
Applying Mitchell definition to another example.

Your e-mail system is monitoring which emails you do (or do not) flag as spam. Based on the collected info, it learns over time how to flag (and filter out) spam e-mails for you.

What is the task T, E, P in this example, among the following?

- classifying emails as spam or not spam T
- watching you label emails as spam or not spam
- the # or fraction of emails correctly classifieds spam/not spam
- none of the above - this is not a ML problem

C.Y.U.



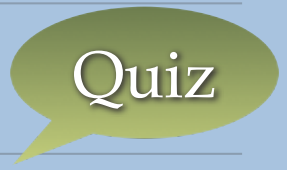
Applying Mitchell definition to another example.

Your e-mail system is monitoring which emails you do (or do not) flag as spam. Based on the collected info, it learns over time how to flag (and filter out) spam e-mails for you.

What is the task T, E, P in this example, among the following?

- classifying emails as spam or not spam T
- watching you label emails as spam or not spam E
- the # or fraction of emails correctly classifieds spam/not spam
- none of the above - this is not a ML problem

C.Y.U.



Applying Mitchell definition to another example.

Your e-mail system is monitoring which emails you do (or do not) flag as spam. Based on the collected info, it learns over time how to flag (and filter out) spam e-mails for you.

What is the task T, E, P in this example, among the following?

- classifying emails as spam or not spam T
- watching you label emails as spam or not spam E
- the # or fraction of emails correctly classifieds spam/not spam P
- none of the above - this is not a ML problem

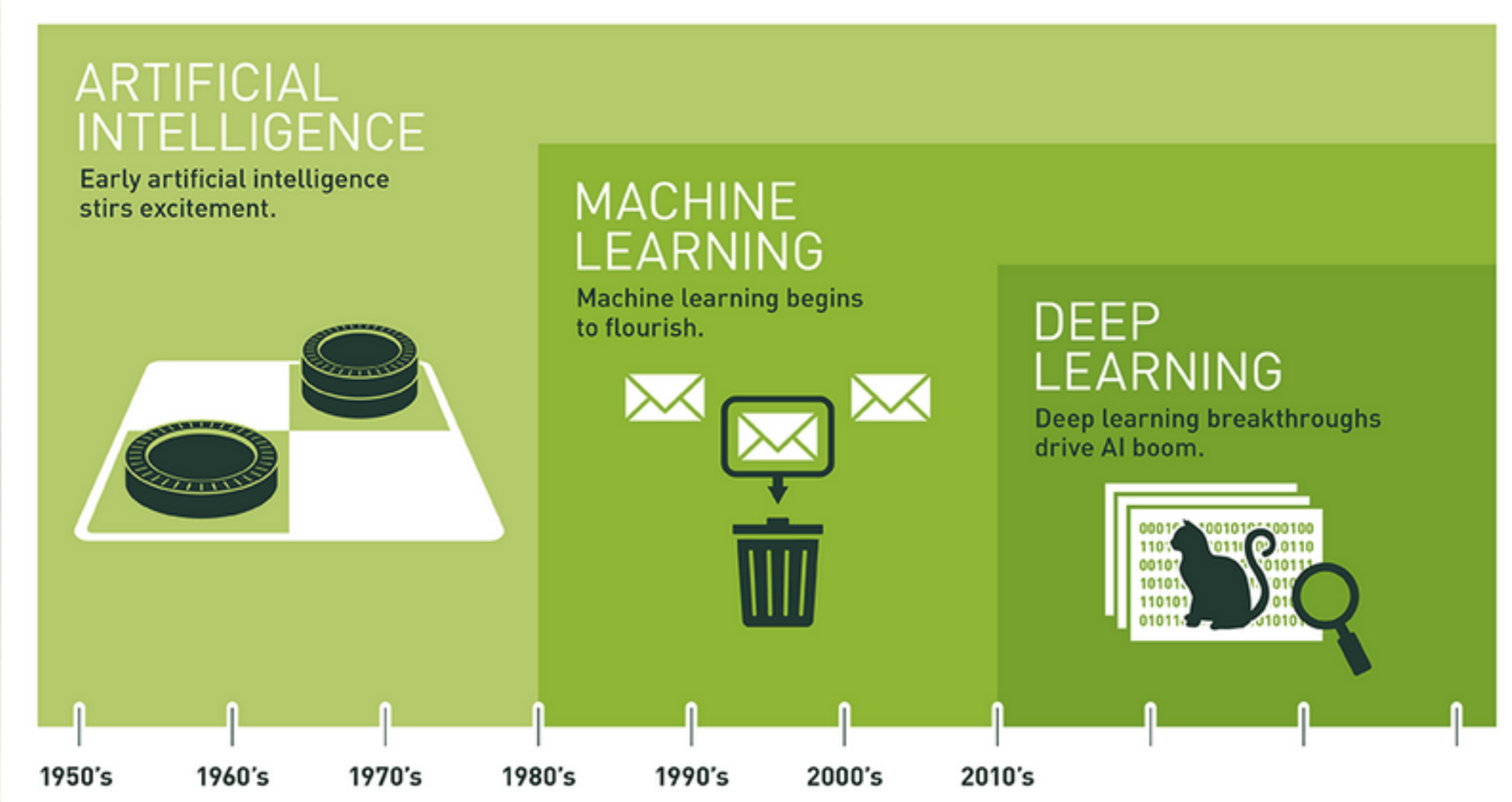
Definition(s) of Machine Learning

"A sub-domain of AI that provides software systems the ability to automatically learn and improve from experience without being explicitly programmed. It relies on an underlying hypothesis about a model one creates, and tries to improve such model by fitting more and more data into the model over time."

– one possible elaboration of definitions from various ML practitioners

Definition(s) of Machine Learning

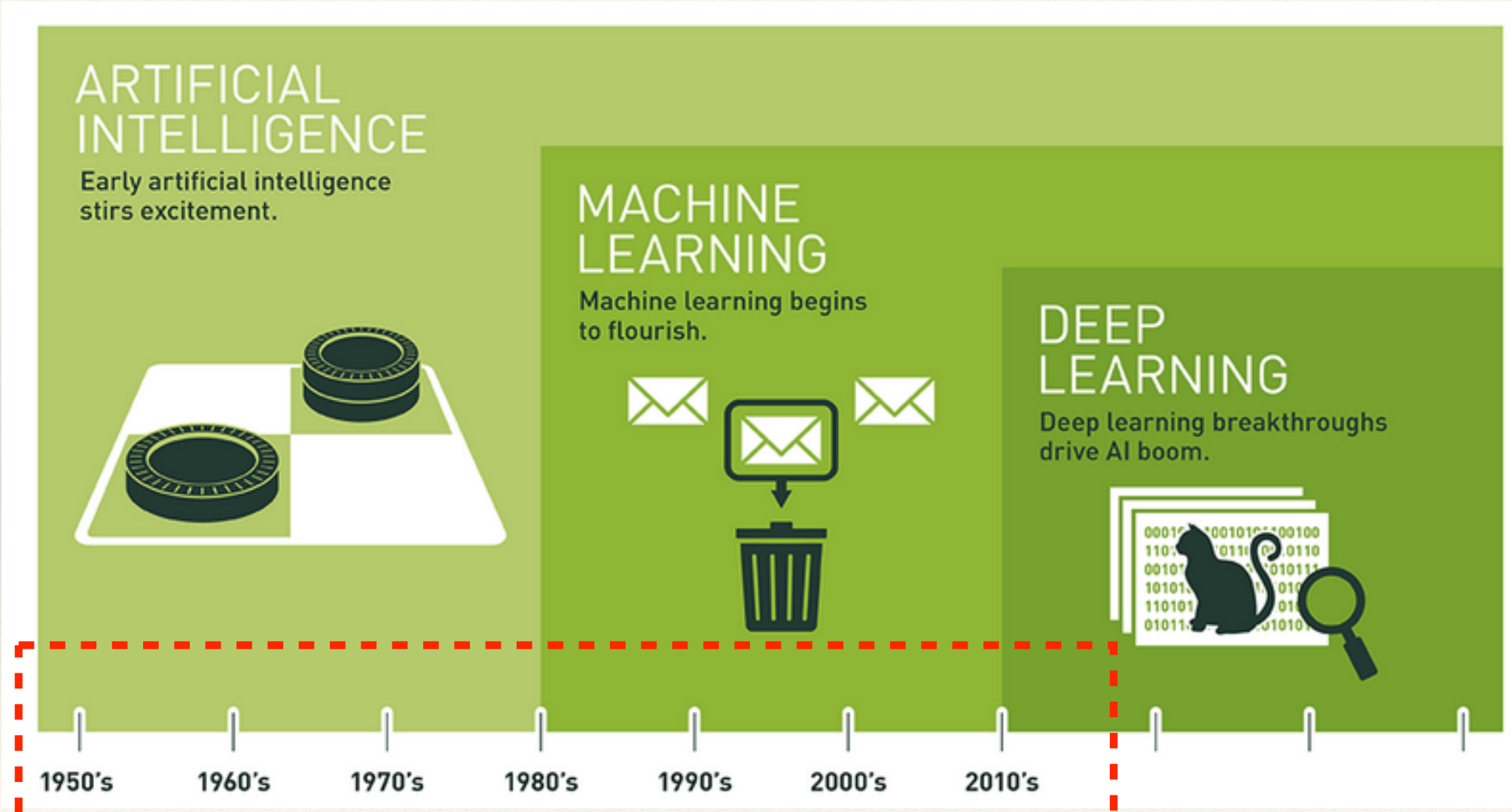
A pictorial definition by a company (Nvidia)



[NVI1]

Definition(s) of Machine Learning

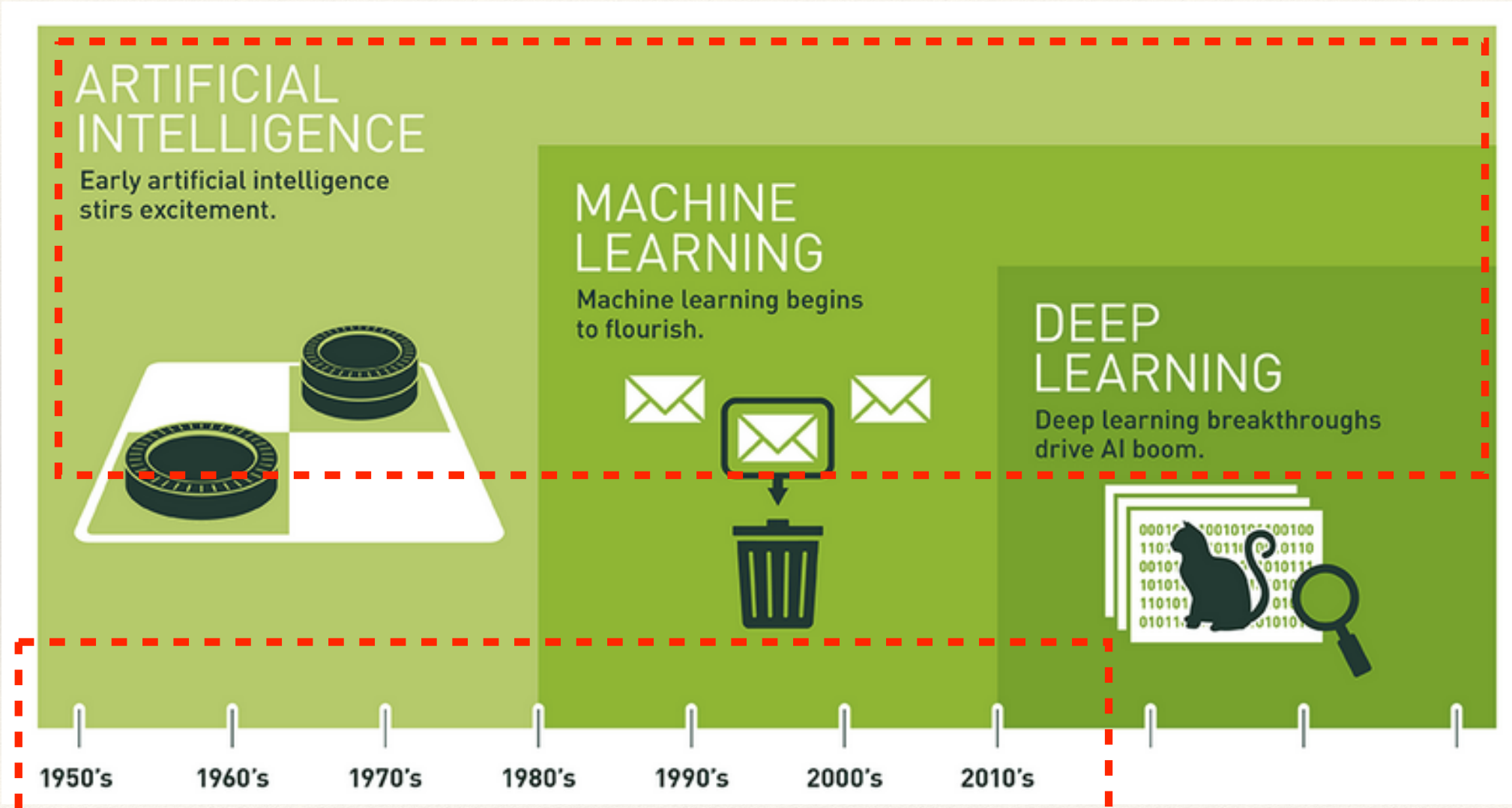
A pictorial definition by a company (Nvidia)



[NVI1]

Definition(s) of Machine Learning

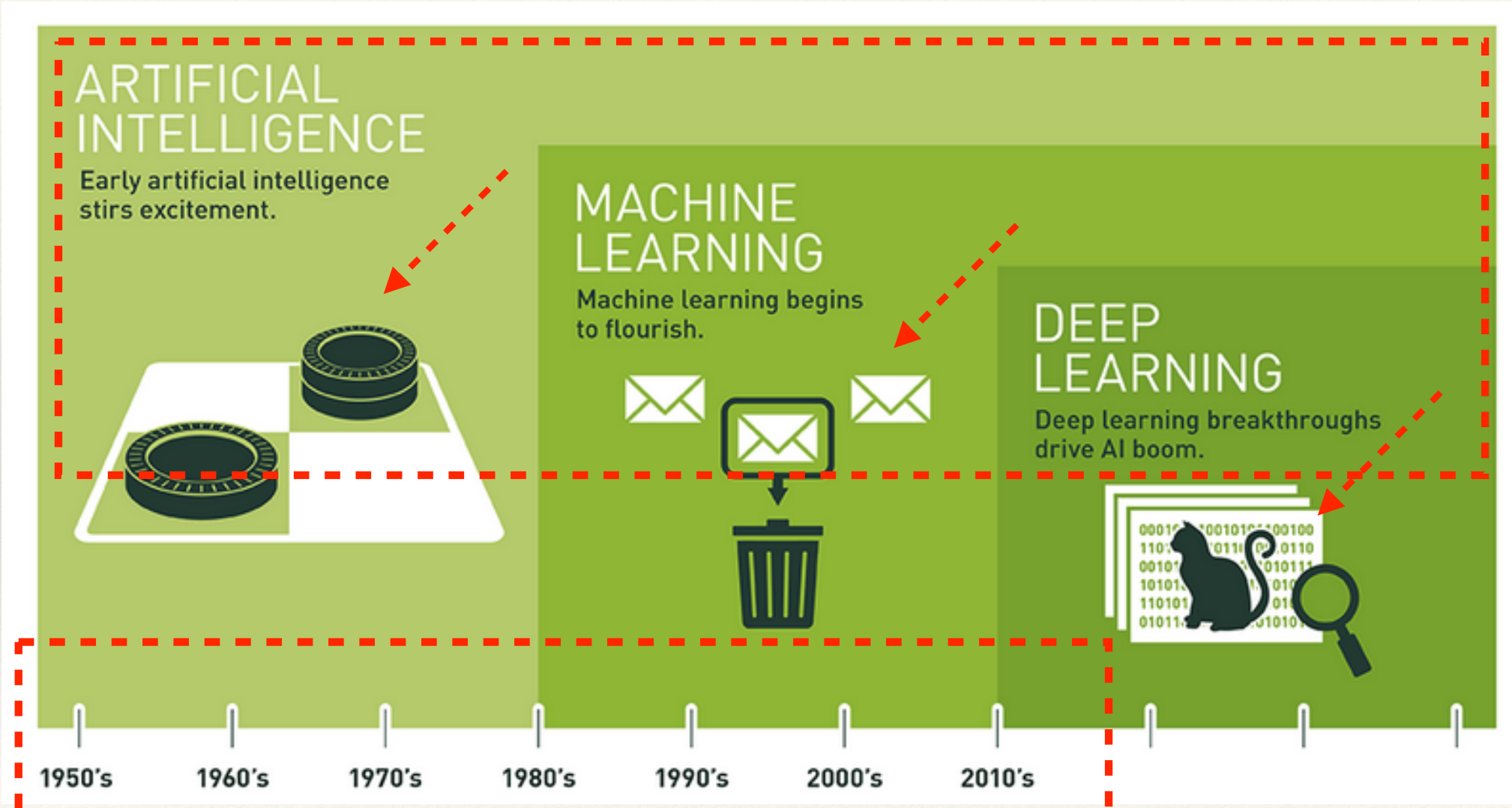
A pictorial definition by a company (Nvidia)



[NVI1]

Definition(s) of Machine Learning

A pictorial definition by a company (Nvidia)



[NVI1]

Learning algorithms are *everywhere*

Not exhaustive list of examples

- **spam** mail filters
- **web search engines** (ranking possible because of a learning algo)
- **clickstream data** collection and consumers' profiling via ML
- **self-customizing programs** (Amazon, Spotify, etc), ML-based recommendation systems, as there is no way to write different programs for million of users
- **pattern recognition**. Handwriting recognition. Natural language processing (NLP). Computer vision.
- It may apply to **Science** (e.g. medical diagnosis applications, computational biology, gene/DNA sequencing, ..), as well as **Industry** (many segments of engineering, predictive maintenance. All fields of research/industry pertaining to finding patterns (at large) and using these to produce predictions is largely influenced by techniques from applied ML.
- …

We will quote and discuss some through the course

“everywhere” → AI vs HI ?

Will AI replace human intelligence (HI)? (“strong AI”, General AI)

- the infamous (boring?) SkyNet quotes on mass media on AI..

No. At least, not in a foreseeable future

- admittedly AI experts think the best approach to implement the ML definition of learning is to try to mimic how the human brain learns, but that’s a different point
- _*Example*_: where is intelligence (HI) located, at the best of today’s knowledge?
- _*Example*_: a baby learning car shapes over time, is this “intelligence”?

Before continuing, we had better reshape our terminology to assess the proper content of the course when we talk about “AI”

- (SPOILER): ʍǝ dɹoqɐqlʎ ɥɐp qǝʇʇǝɹ uoʇ ǝʌǝu nsǝ ʇɥǝ ʇǝɹɯıuoloƃʎ ‚ɐı‚

"AI", really?

"AI" terminology perhaps misleading in most practical discussions

Most of AI research today is actually not trying to recreate intelligence in any shape or form, at all

Artificial Intelligence ?

"AI", really?

"AI" terminology perhaps misleading in most practical discussions

Most of AI research today is actually not trying to recreate intelligence in any shape or form, at all

~~**Artificial Intelligence**~~ → **Automation of task execution** and (eventually) **decision making**

"AI", really?

"AI" terminology perhaps misleading in most practical discussions

Most of AI research today is actually not trying to recreate intelligence in any shape or form, at all

Automation of task execution and (eventually) **decision making**

It is aiming at collecting data around how humans make decisions, to perform the same tasks at a scale (LARGE) and latency (SMALL) that are not humanly possible

- *Example*: facial recognition. The best of us can link thousands of faces to names, without cheating. Machines can do this for hundreds of millions of faces, and in less time. Are such machines smarter than us in a general sense? not even close! But for that specific domain, once trained with large amounts of data, they perform that task at a speed and scale that is beyond what any human may ever hope for.

This is the "artificial intelligence" we are talking about.

The ML road to "*everywhere*"

So, all started from AI decades ago..

- original idea was to build intelligent machines by programming them

.. but programming for a task work for *some* tasks - not for *all* tasks…

- we can program a machine to e.g. solve math, or find the shortest path from A to B
- we cannot program a machine to read-and-tag a spam email or to see-and-tag a photo - unless we accept to continuously retune the algorithm, personalise it, etc

.. so, it soon became clear that the only way to perform such tasks is to **enable a machine to learn how to perform specific tasks by itself**.

Integration of intelligences

Areas that may benefit from this "automation of task execution and decision making" are areas where such advancement would eventually **help humans**.

In this scope, it is beneficial to use machines instead of humans in tasks that either humans do not want to do or cannot do as well

- *Example*: AI-based systems able to retain high-performances also in high-stress conditions or environmental dangers unbearable by humans

The keyword might be **integration**. I.e. AI "**together with**" HI, and not "**instead of**".

- *Example*: recent progresses in ML-based radiology, yielding unprecedented results, as well as surprisingly interesting ethical implications in not using AI..

Why now?

A crucial question: **if all this was already available decades ago, why the rise of AI we see today has not happened earlier?**

Acceleration towards larger adoption of ML

A revive and acceleration happened recently, mainly because of factors that I would list as:

- the raise of **Big Data**
- the **technology** progresses (e.g. GPUs)
- "Democratisation" [*] of massive computing resources via **Cloud** approaches

For DSC students: this is largely covered in another course (on Cloud and Big Data)

[*] as most of these resources world-wide are far from being free-of-charge, and Big Data companies implement carefully designed business models, this is debatably "democratic" in a social sense. Not discussed here further, though: here we aim at stating that "in principle" you have a "pay-and-access" option on not-on-premise resources, which did not exist before.

Today, it is a fact that ML/DL are among the core transformative technologies at the basis of most world-wide activities aiming at extracting **actionable insight from (big) data**.

Breathing time..

We better explained what we mean with AI in the scope of the course, and hence set the context in which this course will live.

Questions so far?

How do we approach Applied ML **in this course**

A standard approach:

- theoretical foundations → rigorous application to a problem → solution
  - implies all math and statistics, state-of-the-art in ML algos, …

Our approach:

- problem class definition → intuition of theory needed → adoption of an adequate tool to implement theory → application of best practices → solution

Largely resumable in "artisan" kind of work..

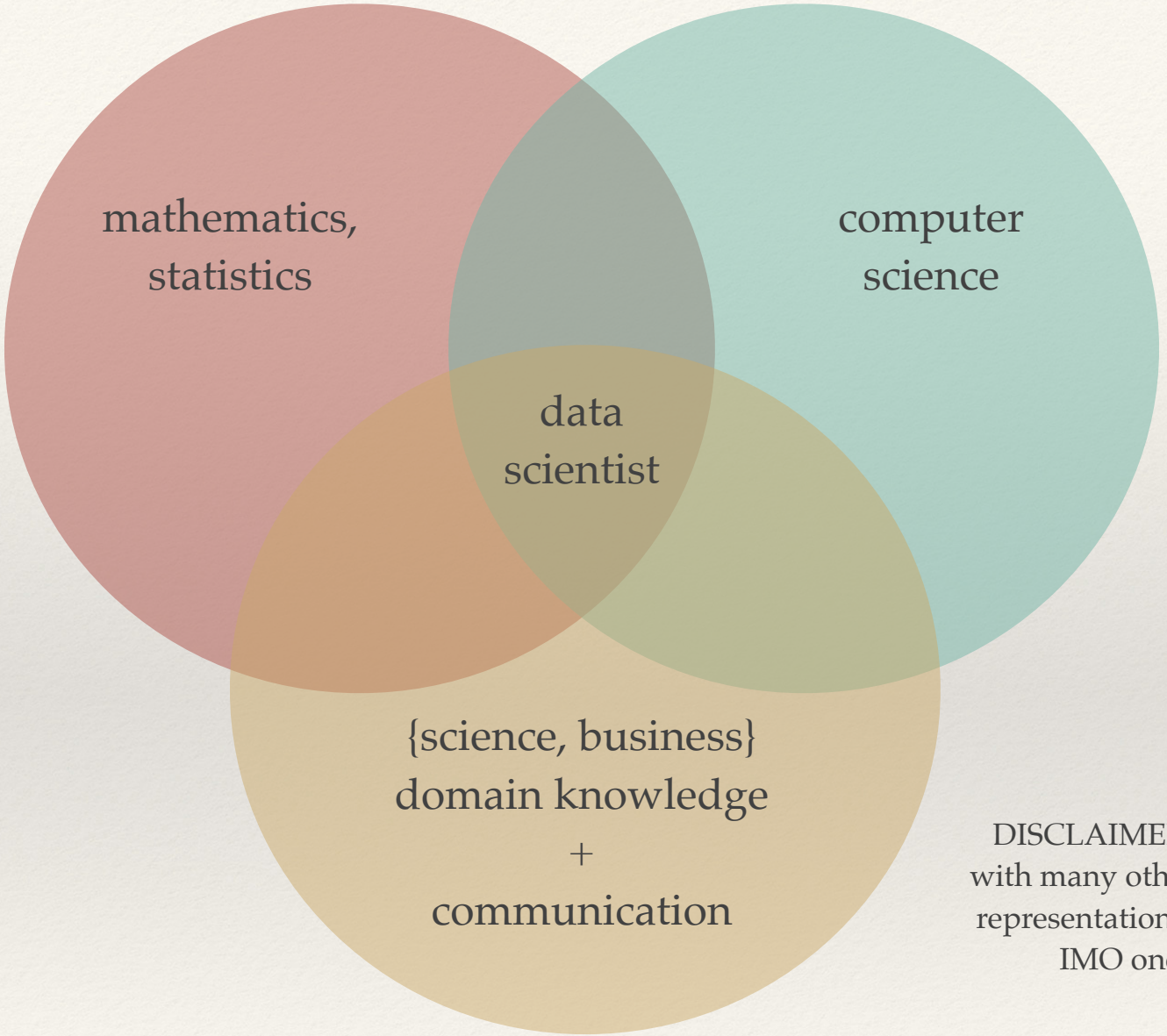
Which is basically what a data scientist does.

Oh, BTW, **what is data science**?

Quite hyped..


Harvard Business Review
GETTING CONTROL OF BIG DATA
Harvard Business Review
DATA SCIENTIST
The Sexiest Job of the 21st Century

What is a data scientist?



DISCLAIMER: I would disagree with many other ways this pictorial representation is displayed. This is IMO one of the fairest.

Not uncommon that a data scientist is referred to as "a unicorn"…

Best definition?

“A data scientist does not exist, but a data science team does” ©

(.. Should I care? **Yes**. Many other people will anyway see you as such. Some of you are even enrolled in a PhD program that has “data science” in the name.. You had better define yourself before others do.)

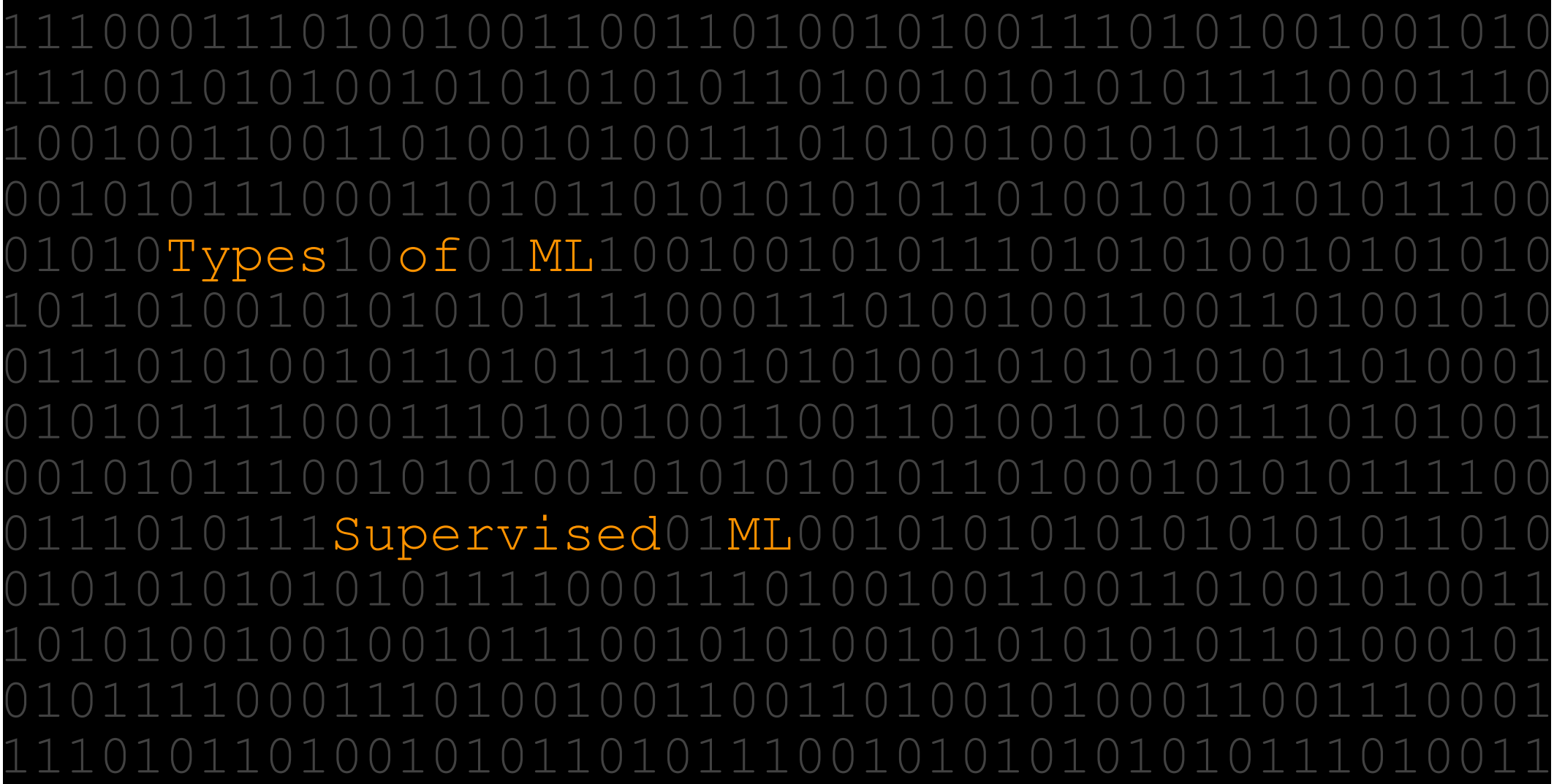

Types of ML

Types of ML

According to traditional classification, the 3 most populated classes of learning algorithms are:

- **supervised** learning: teach the machine how to do something
- **unsupervised** learning: let the machine learn by itself what to do
- **reinforcement** learning: make the machine learn by feedback

In this part of the course, we will do mostly supervised, some unsupervised, and on reinforcement learning.. we will see!


Types of ML
Supervised ML

Supervised ML by examples

Example: we want to predict housing prices in an area.

- good example of a ML project that starts from a need
  - e.g. you have your house to sell, you have area data, you have a clear goal

A learning algo could:

- fit a straight line to the data and give you a predicted sell value

A better learning algo could:

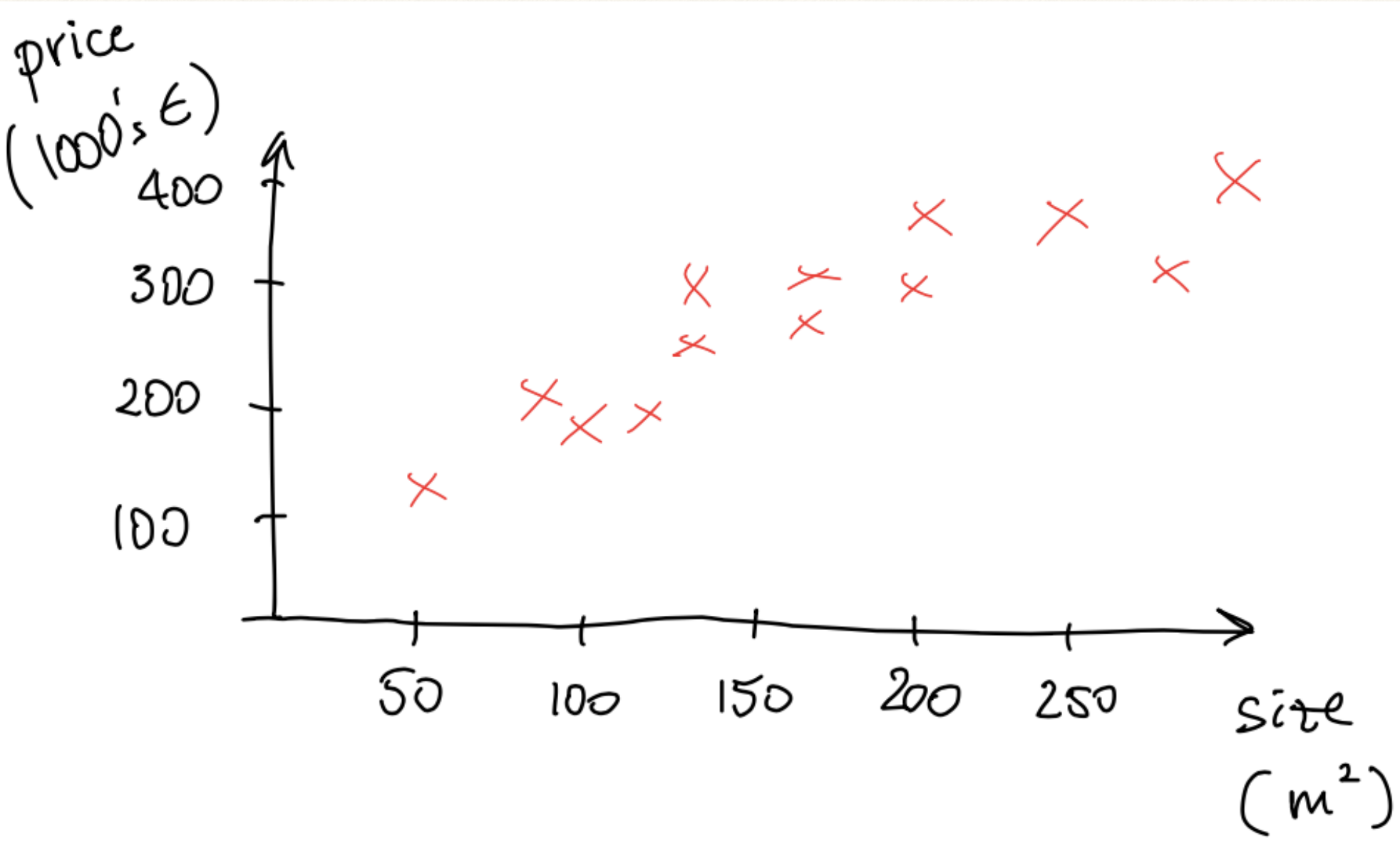
- find better ways: e.g. a second-order polynomial fit to this same data?

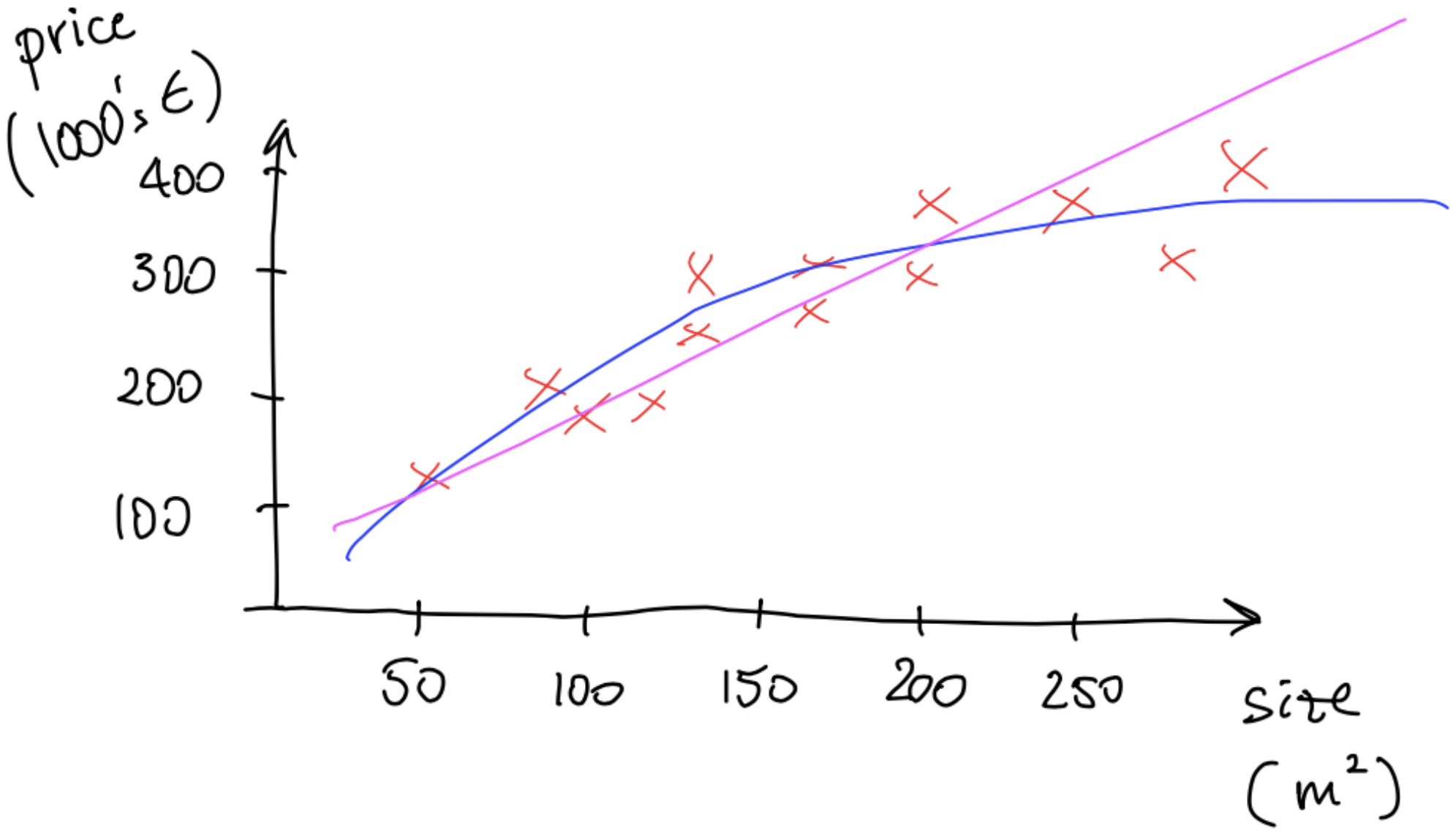
Both are "**supervised**" learning algorithm

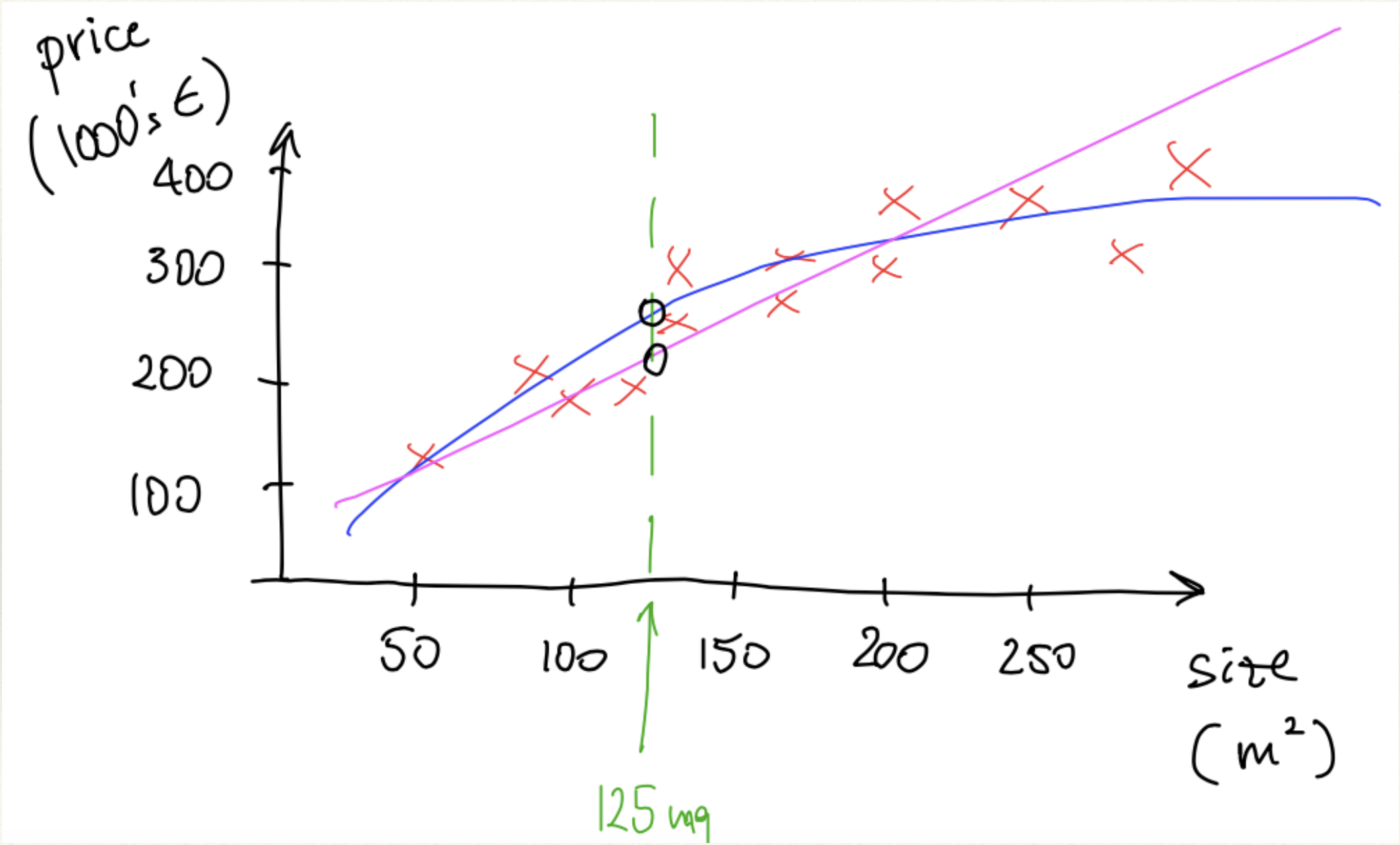
- we gave the algos a dataset (e.g. location, size, .. and prices) in which the "right answers" (prices) were included.

This is a "**regression**" problem example

- i.e. we are trying to predict a continuous value output (namely, a price)


price
(1000's €)
400
300
200
100
50
100
150
200
250
size
(m²)


price
(1000's €)
400
300
200
100
50
100
150
200
250
size
(m²)


price
(1000's €)
400
300
200
100
50
100
150
200
250
size
(m²)
125 sq

This example

The housing example was a **supervised-ML regression** example.

Another example

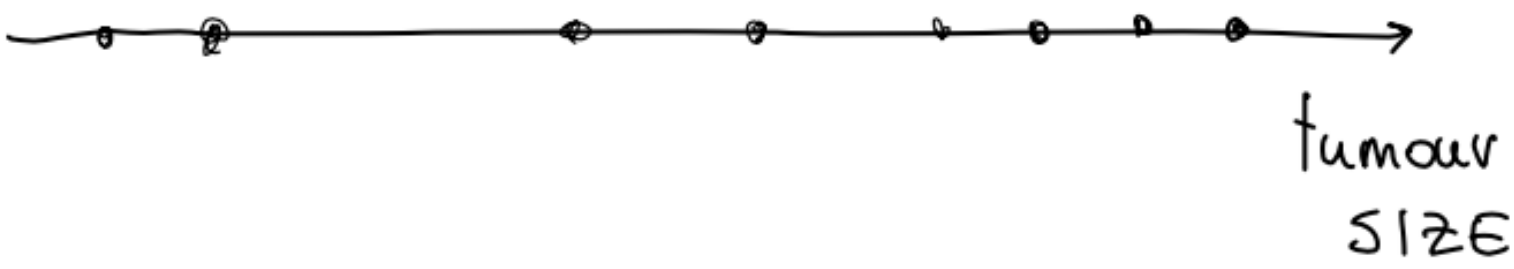
Example: Medical records, you have data on breast cancer size. Goal is to try to predict if a given breast cancer is malignant or benign.

- suppose you have an additional, new patient with a tumour of a given size: can ML use the available data to estimate what is the probability that this tumour is malignant versus benign?

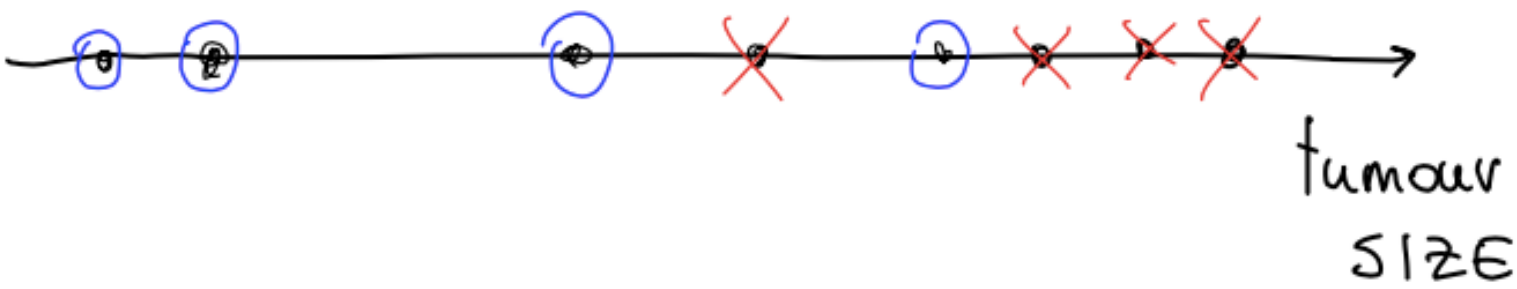
This is an example of a "**supervised-ML classification**" problem

- we are trying to predict a discrete value output (namely a 0 or 1, benign or malignant)

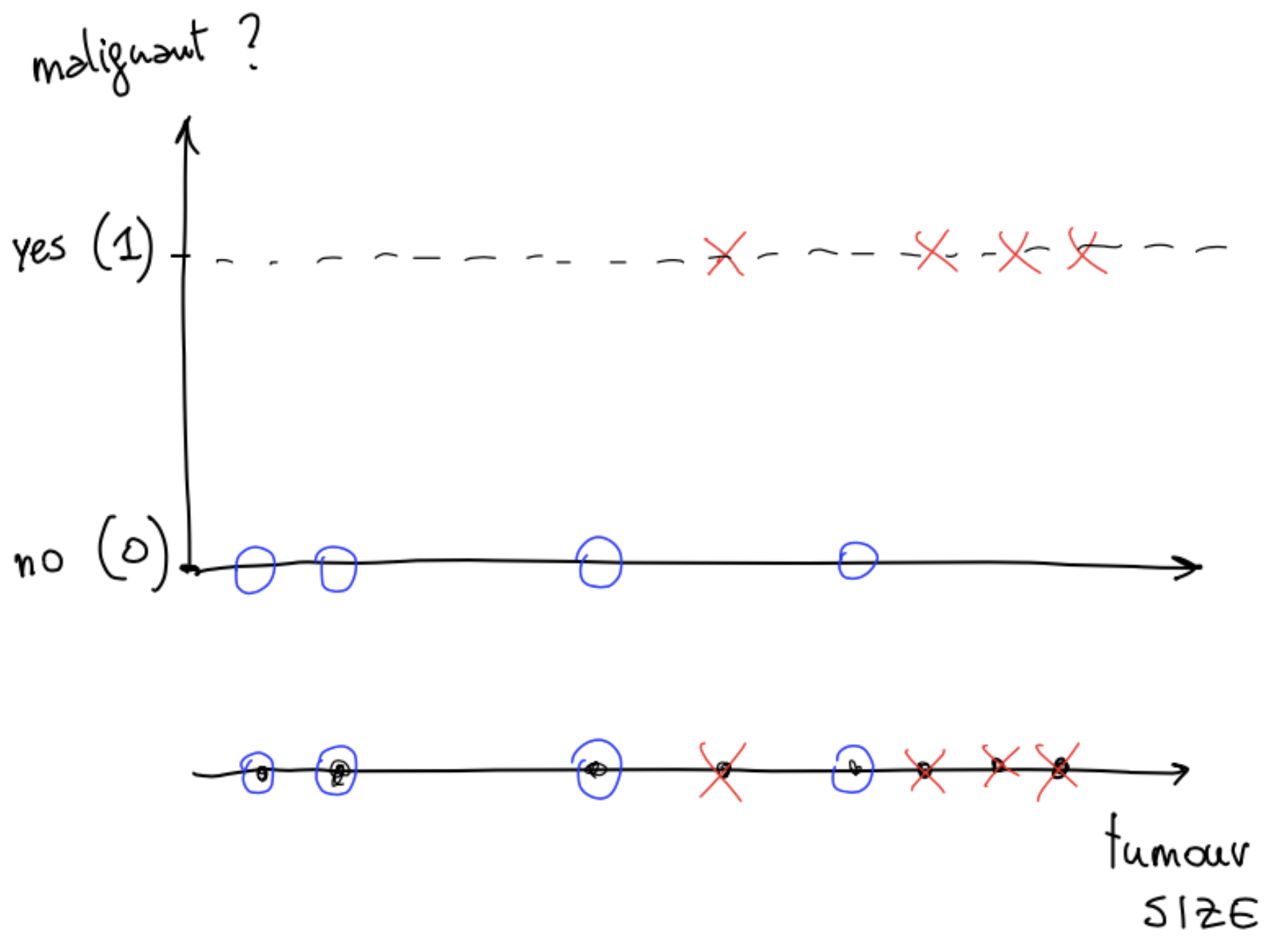
If you have only the tumour size..



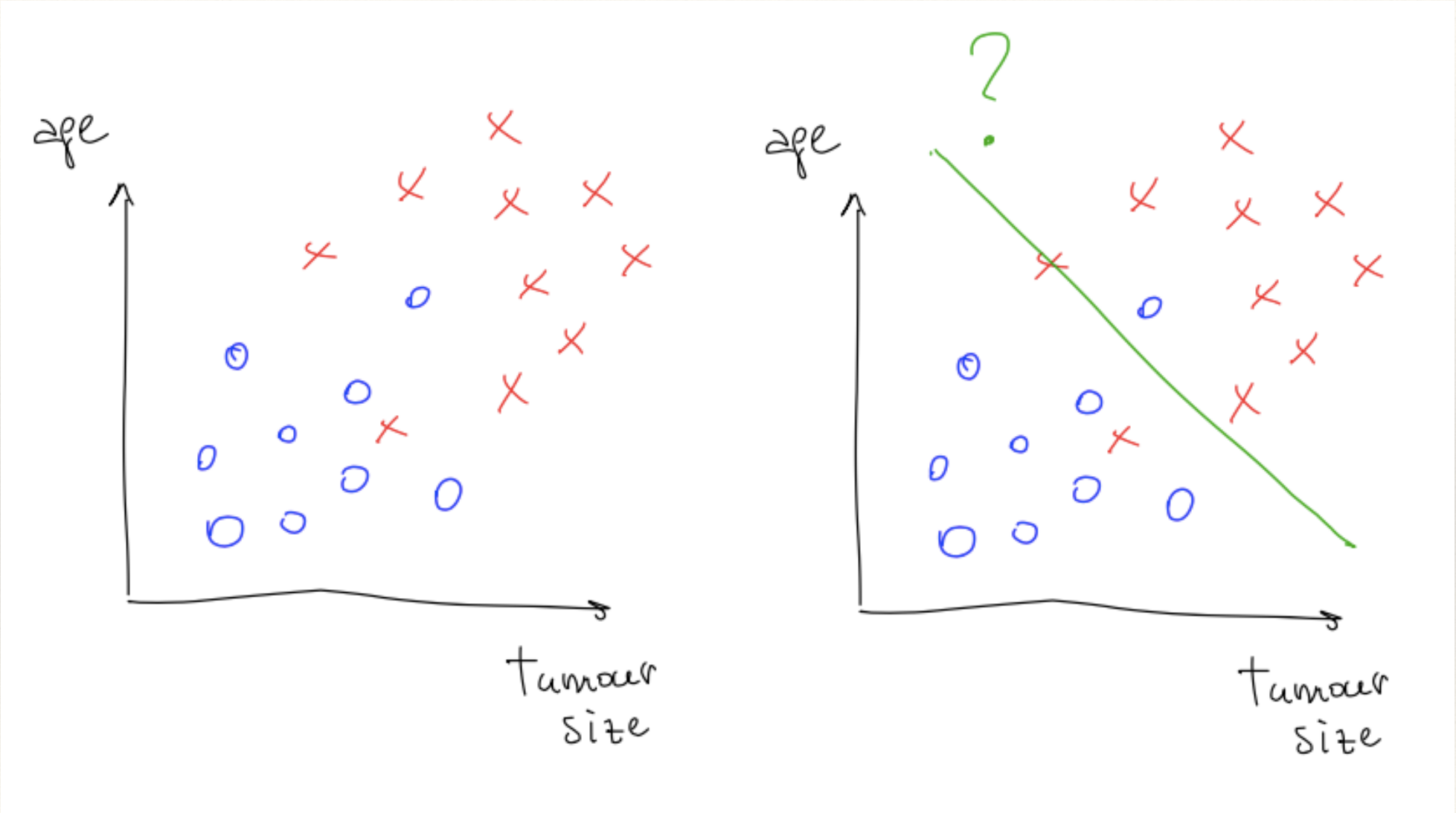
If you have only the tumour size..



If you have only the tumour size..



If you have the tumour size and the age..



If you have size, age, … → Big Data

In words: if you have a larger, better, more diverse dataset…

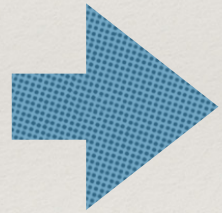
- i.e. not only the tumour size but also e.g. the age of the patient, or the tumour thickness, info on uniformity of cell size, or uniformity of cell shape, …

.. a ML algo can try to make use of all data "parameters" (we will pick a better name soon) and use them to crunch predictions

- up to an extremely large number of such parameters..

Wow. Do you see the potential?

Ready for Quiz 1



Take-away message

In **supervised ML**, we have a dataset, plus some idea that there must be a relationship between the input(s) and the output. And we already know what our correct output should be.

Supervised ML problems can be categorised into "**regression**" and "**classification**" problems.

In a regression problem, we are trying to predict results within a continuous output

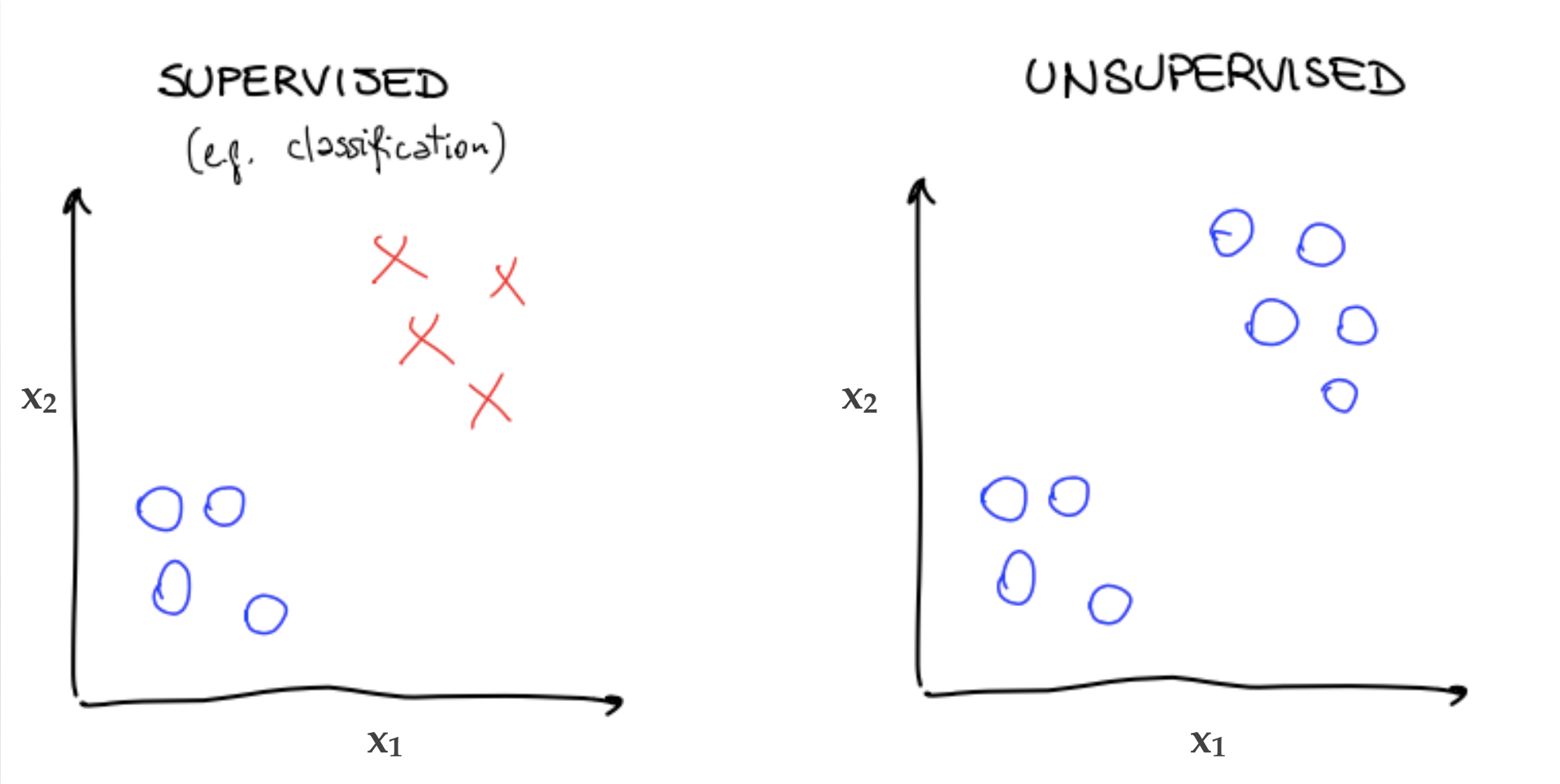
- i.e. trying to map input variables to some continuous function

In a classification problem, we are instead trying to predict results among a discrete output set

- i.e. trying to map input variables to some discrete categories


Types of ML
Unsupervised ML

Supervised vs Unsupervised



we knew positive and negative examples, we were told explicitly what is the right answer

all data has the same "**label**", i.e. no label at all

Unsupervised learning

In unsupervised ML, we are admittedly a bit more blind.

We have the data, sure, but we are *not* given any "**label**", so we do not know which class each data point belongs to.

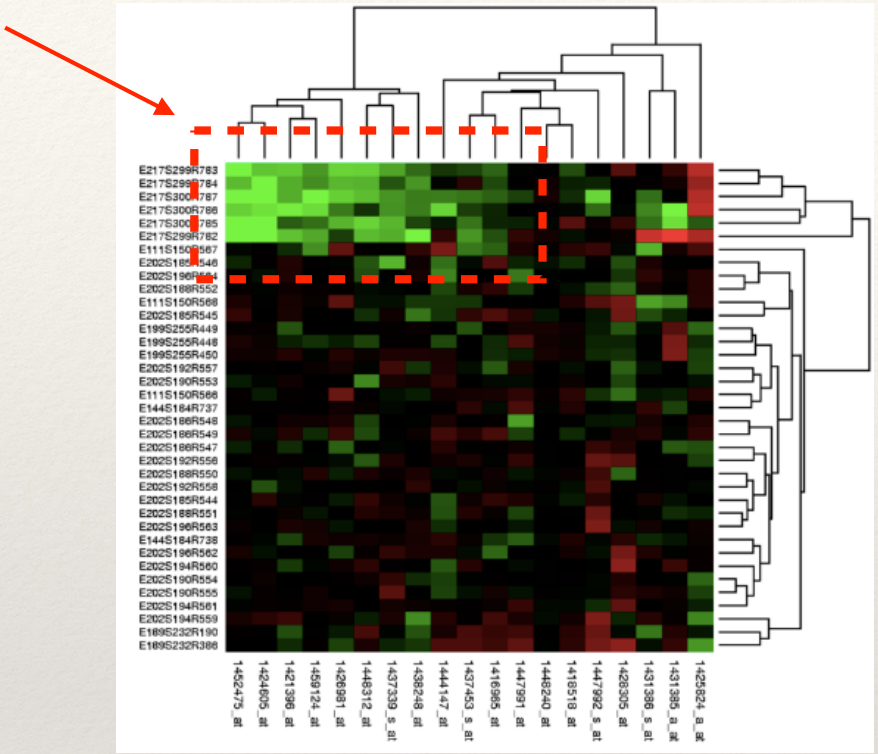
We are just told "***can you find yourself some structure in the data?***"

- i.e. "I am not supervising you in this task, can you nevertheless perform it?"

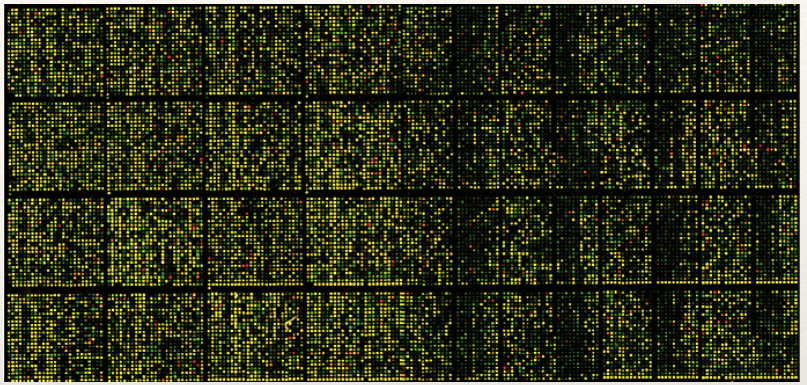
See last slide: an unsupervised ML algo might decide that the data lives in 2 different clusters: this is a so-called "*clustering algorithm*"

- *Example*: news glueing into cohesive stories containers online (e.g. Google News), all based on automatic clustering algos

Another example: understanding genomics



[DNA1]



Genomic20K - Image Genomic Image of the Core's 20k human Oligonucleotide microarray comparing MCF10 & UACC1179 breast cell line expression profile) [DNA2]

Example of **DNA microarray data**, or similar.
One measures how much (the colours) certain genes are "expressed" in a group of patients.

[DISCLAIMER: approximate explanatory description, not intended to be scientifically correct]

Unsupervised ML with Big Data

We are not given any label here, by construction ("**unlabelled data**")

- we know nothing about these people (e.g. age, gender, ethnic group, pre-existing conditions, ..)

We can think of running a clustering algorithm and - based on this data - to group individuals into different categories (e.g. "cat 1", "cat 2", ..)

- NOTE: *a-priori a data scientist does not know what each category might mean* (so, my lack of domain knowledge here is acceptable..)

We can run algos to use this data (that we are given) to group people (that we know nothing about) into groups (that were not a-priori defined by us).

Everything is decided by just data and algos. This is **unsupervised ML**.

- already at this stage, you probably already developed some intuition as of why this is so important in the **Big Data** era, in terms of an ability to get insight from vast amounts of unlabelled data

Other examples of unsupervised ML

Example: used to **organise large computer clusters**, trying to figure out which machines tend to "work together": if set-up takes this into account, the data centre works more efficiently.

Example: **social network clustering**. Given knowledge about which friends you message the most, or given your <pick-your-social> connections, try to automatically identify which are cohesive groups of people who know each other (or may want to connect).

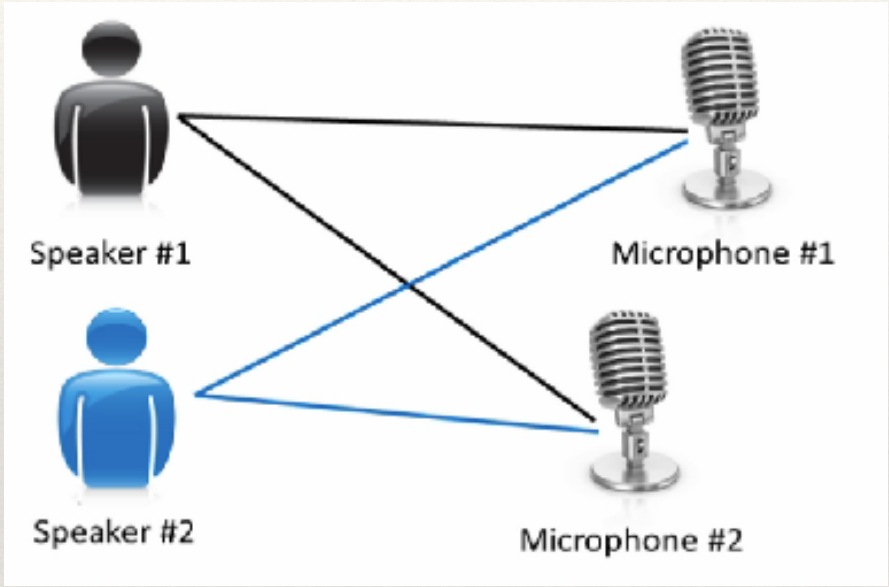
Example: **Market segmentation**. Analyse huge DBs of customers' info and try to automatically group customers into different market segments, to target advertisement, offers, etc.

Example: e.g. **Physics/Astronomy data analysis**. Clustering algos might give insight into possible logical grouping of previously disconnected data.

Not only clustering

Example: the cocktail party problem. It is representative of the (large) signal processing category of problems, for example.

Machines are terrible at this, while humans perform very well, and easily



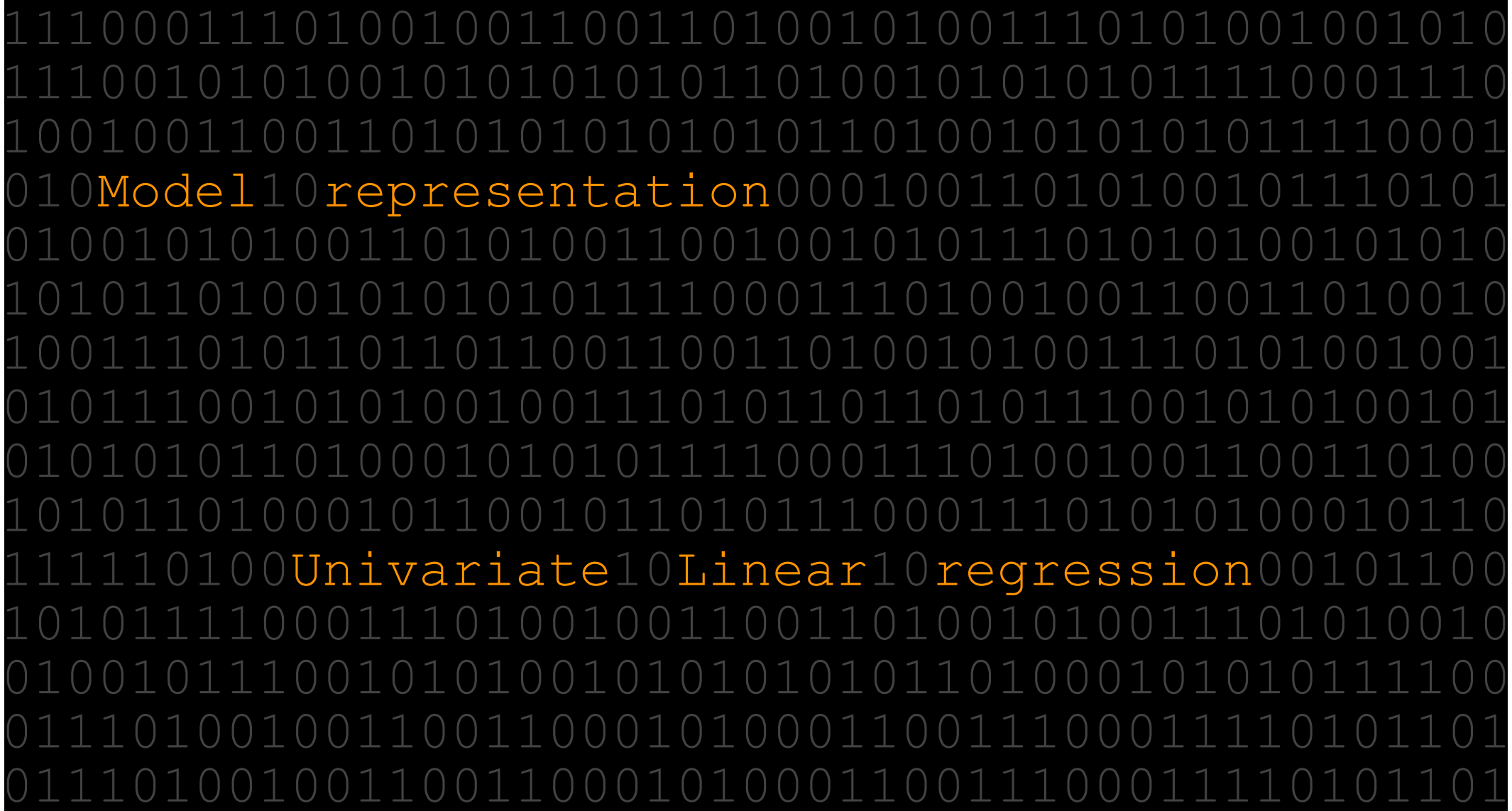
[COP1]

This is addressed as a **unsupervised** (non-clustering) **learning** problem: goal is to **find structure in a chaotic environment**.

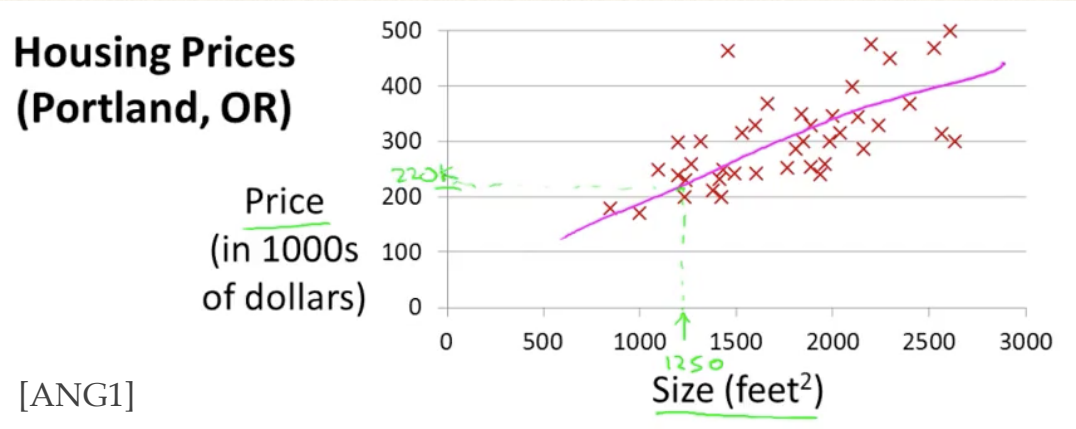
Summary

Unsupervised ML allows to approach problems with little or no idea about what results should look like. We *derive structure from data*, we do not necessarily know the effect of the variables, there is no feedback based on the prediction results.

We can derive this structure by *clustering* algos (i.e. grouping the data based on relationships among its variables) or *non-clustering* algos (i.e. find structure in chaotic environment).


Model representation
Univariate Linear regression

Let's use this dataset:



[ANG1]

It is **supervised learning**

- because we're given the "right answer" for each example (house details, and price it was sold at)

it is **regression**

- we are predicting a real-valued output, namely the price

You can easily draw a reasonable line, no?

- NOTE: I am trying hard not to say the word "fit" :)

(one remark before proceeding)

I know all of you are familiar with linear fit, and so on and so forth.

But in **applied** data science and **applied**-ML/DL my view is that one needs to become confident with an often less formal and more pragmatic approach, plus a somehow more generalised and slightly different terminology and way of thinking.

[DISCLAIMER: not everyone agrees on this argument. You are encouraged to read around and form your opinion!]

Notation and terminology

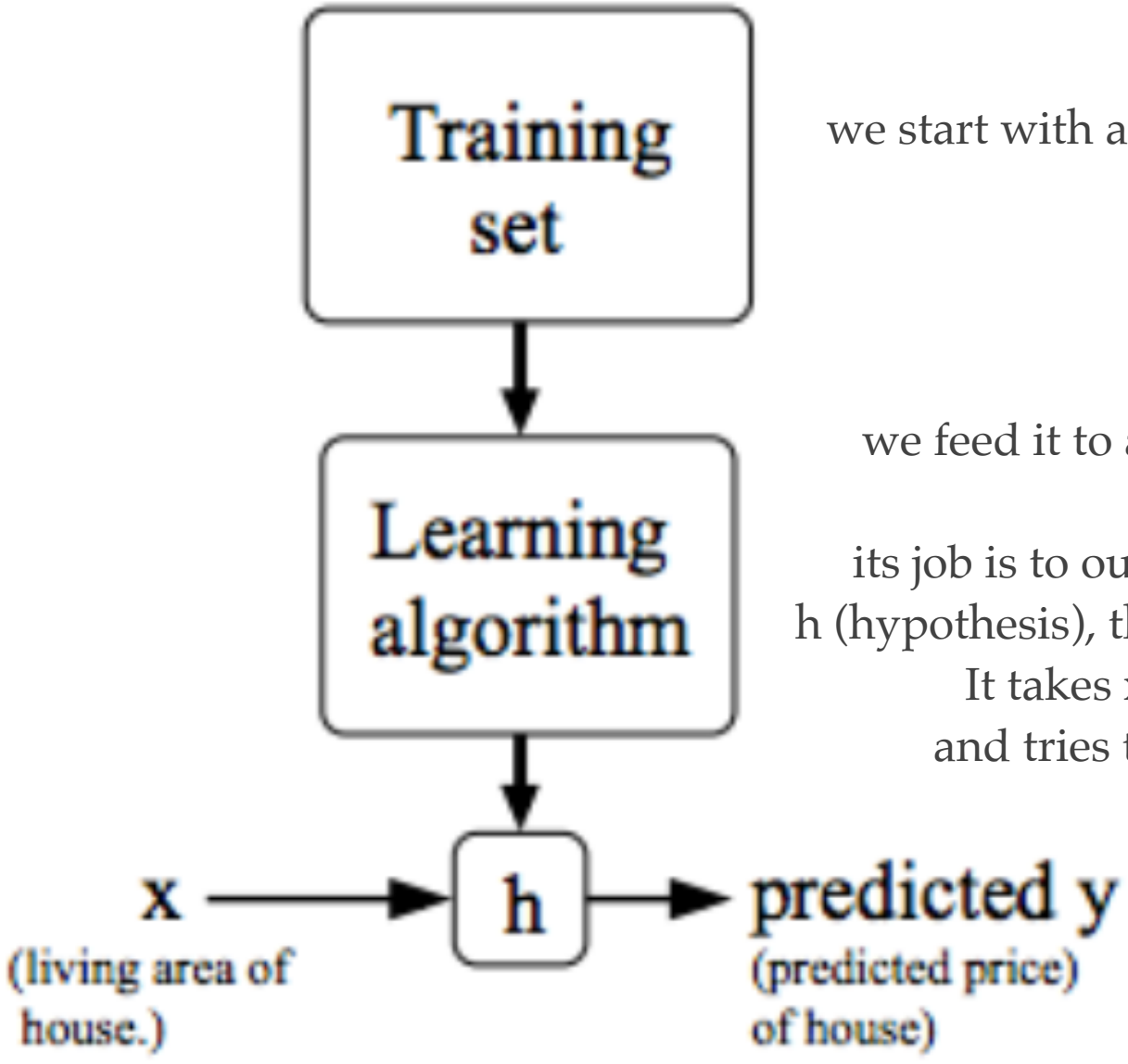
Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

this housing **dataset** will be called our **training set**

Notation (better to familiarise with this early enough):

m	# of training **examples** in the **training set**
x	"input" variables, or "**features**"
y	"output" variable, or "**target**"
(x, y)	a single training example (a row in the table above)
$(x^{(i)}, y^{(i)})$	i^{th} training example (i^{th} row)

ML "modelling"



we start with a training set

we feed it to a learning algo

its job is to output a function
h (hypothesis), that maps x onto y.
It takes x as input
and tries to predict y

[ANG1]

Hypothesis

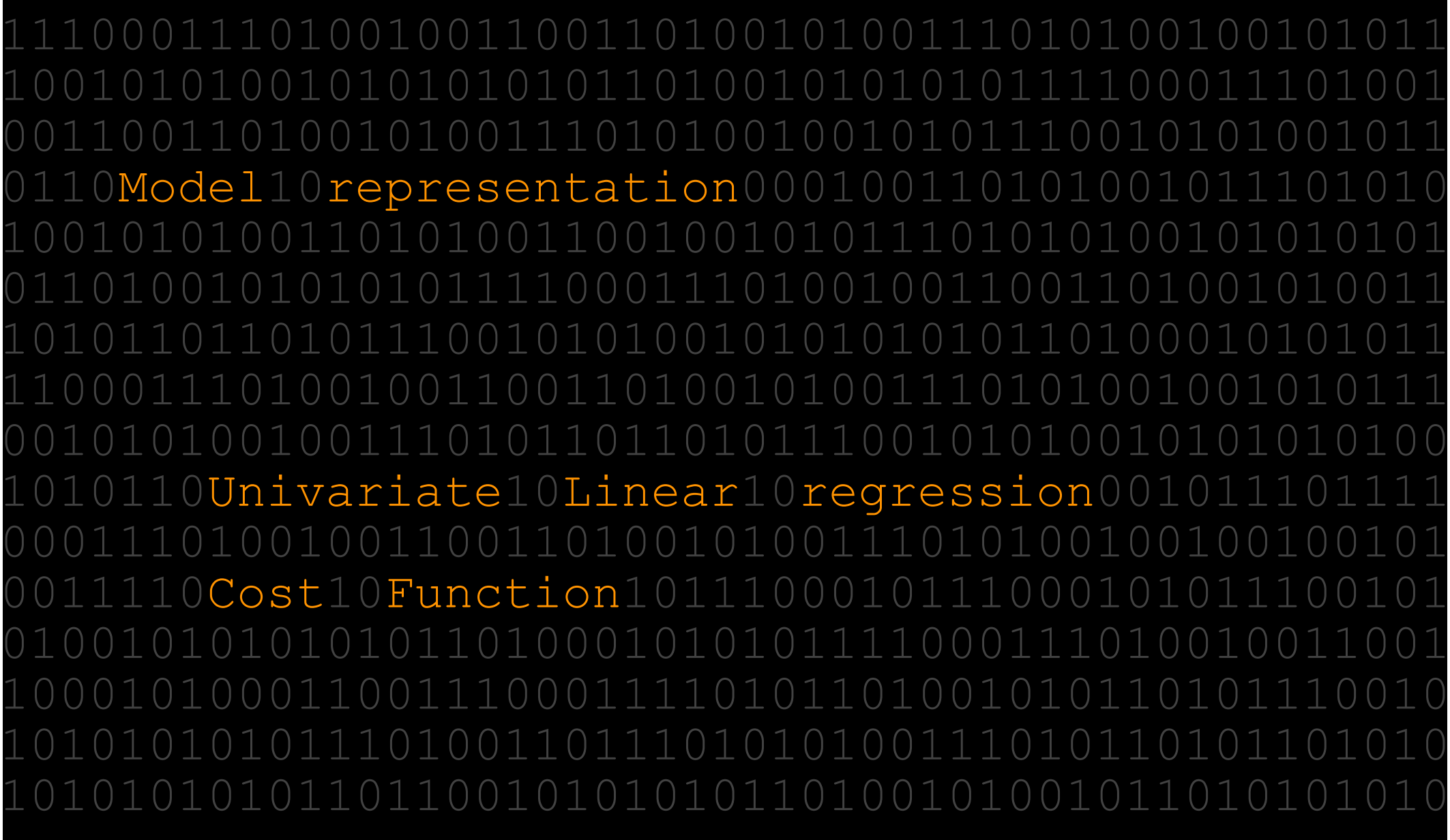
How do I represent this hypothesis h? $h_\theta(x) = \theta_0 + \theta_1 x$

θ are called "**parameters**" of the model

- so, we are making the hypothesis that y is a linear function of x
- *why linear?*
  - sometimes we will want to fit more complicated, non-linear functions as well. But the linear case is the simplest building block to start with. We will complicate this, and eventually build more complex learning algos.

This ML model is called **univariate linear regression**. Meaning?

- "linear" and "regression": obvious by now..
- "univariate": i.e. based on one variable (x) only, i.e. we are predicting all the prices as functions of just one variable (x)


Model representation
Univariate Linear regression
Cost Function

Recap

Training set

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

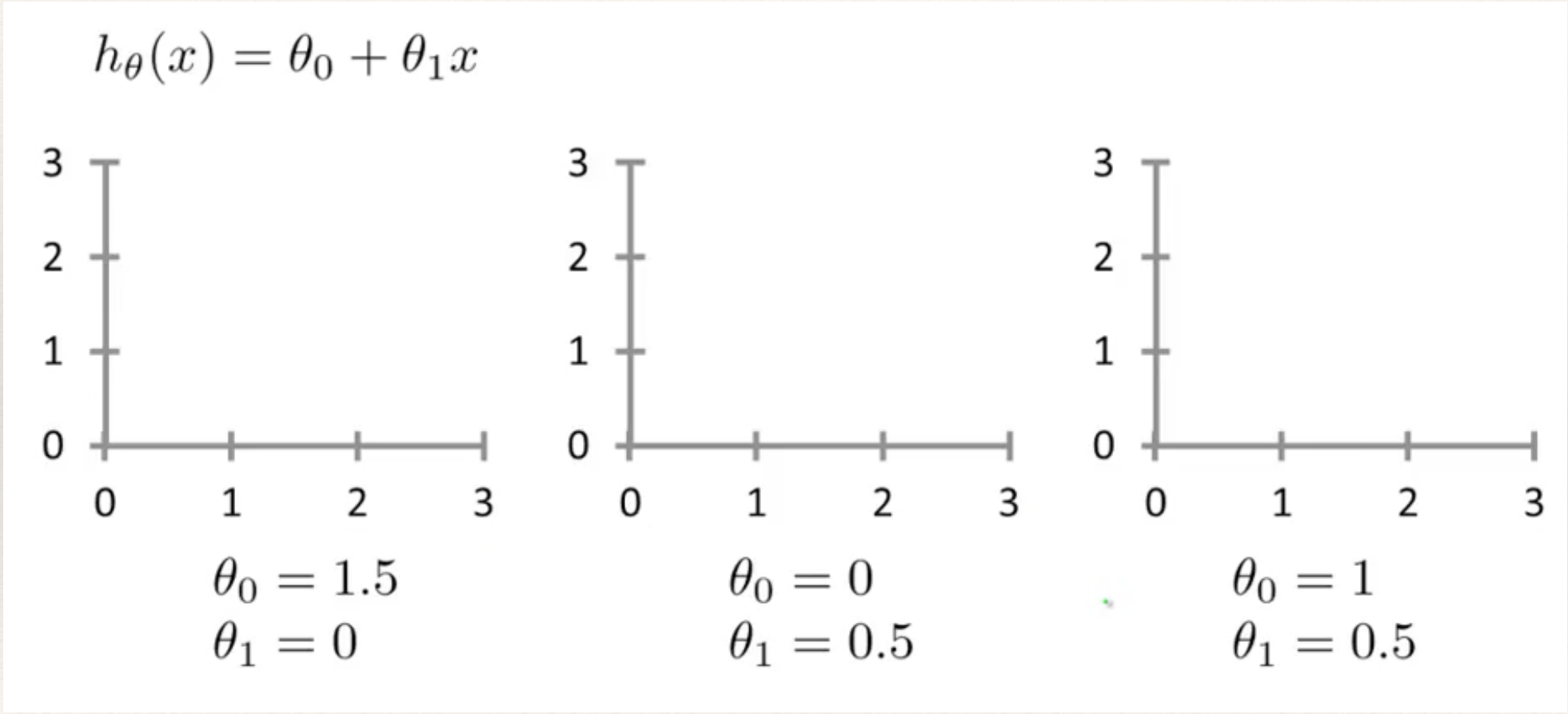
m

Hypothesis $$h_\theta(x) = \theta_0 + \theta_1 x$$

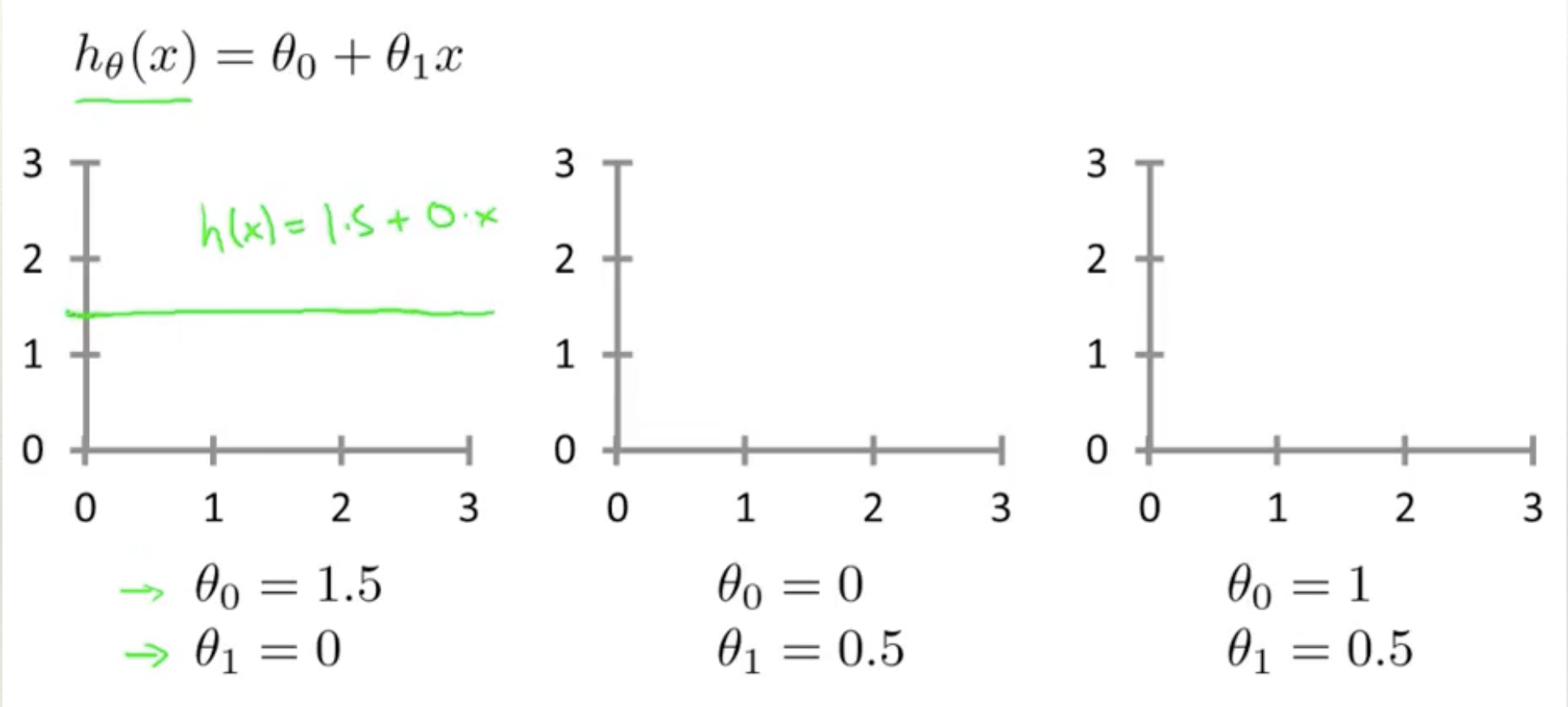
Parameters $$\theta_0, \theta_1$$

How to choose the parameters θ?

We can make different choices of the parameters…

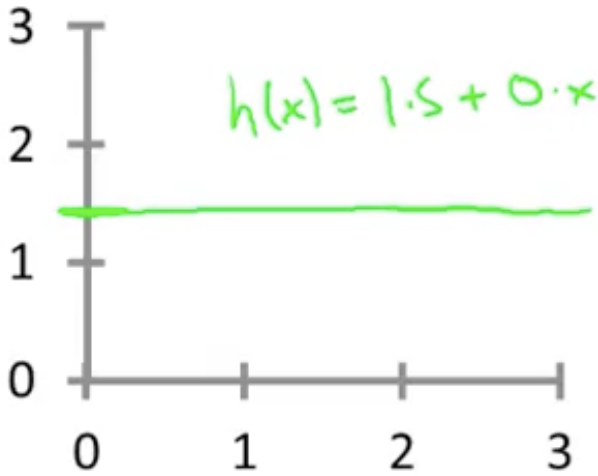


[ANG1]


$h_\theta(x) = \theta_0 + \theta_1 x$
$h(x) = 1.5 + 0 \cdot x$
3
2
1
0
0 1 2 3
$\theta_0 = 1.5$
$\theta_1 = 0$
$\theta_0 = 0$
$\theta_1 = 0.5$
$\theta_0 = 1$
$\theta_1 = 0.5$

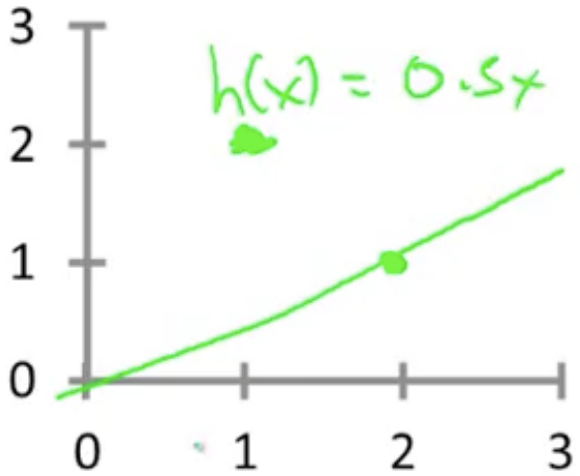
[ANG1]

$h_\theta(x) = \theta_0 + \theta_1 x$



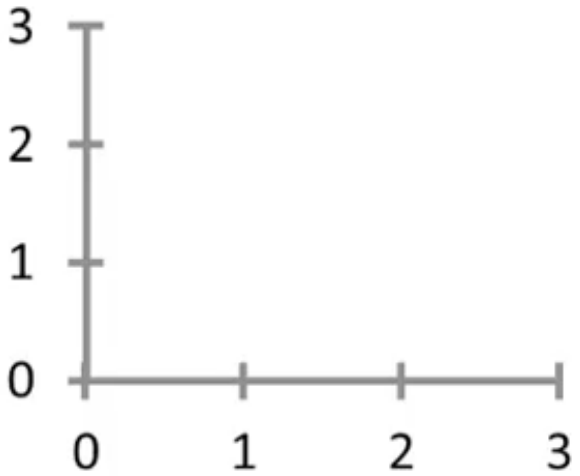
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

$\theta_1 = 0.5$

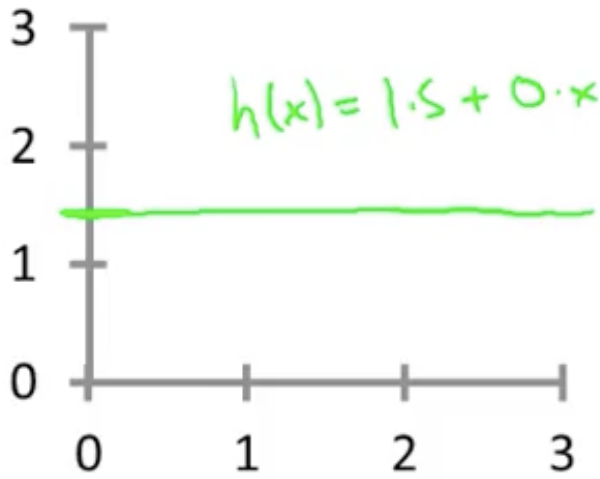


$\theta_0 = 1$

$\theta_1 = 0.5$

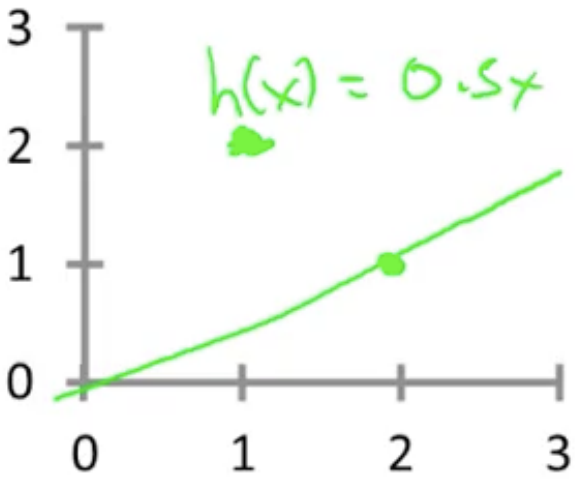
[ANG1]

$h_\theta(x) = \theta_0 + \theta_1 x$



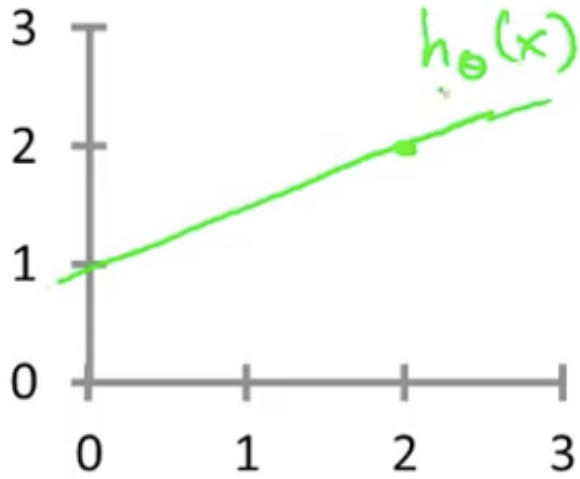
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

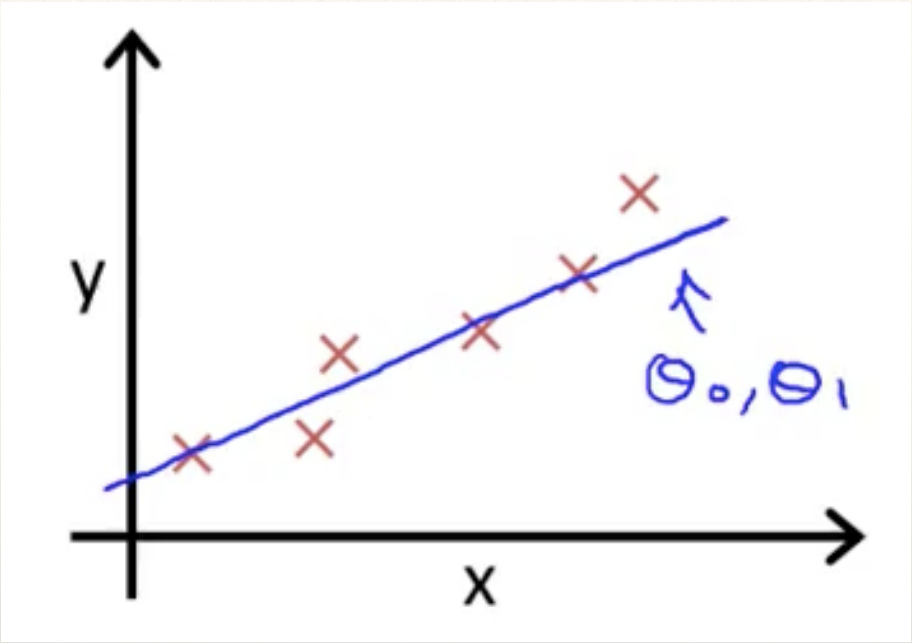
$\theta_1 = 0.5$



$\theta_0 = 1$

$\theta_1 = 0.5$

[ANG1]



[ANG1]

How to "fit" the best possible straight line to our data?

- how do I choose the θs?

The idea is that we should choose our parameters θ_0 and θ_1 so that **h(x)**, i.e. the values we predict for the outcome of inputs x, is close to the "true" values y for the examples of the type $\mathbf{(x^{(i)}, y^{(i)})}$ in our training set

- i.e. let's try to make this true at least for the training set

Minimisation

Let's formalise this in a way that is pragmatic for applied ML.

I want to solve a **minimisation problem**:

- I want to minimise over θ_0 and θ_1 the difference between **h(x)** and **y**
- take the squares of such differences (e.g. I do not need to care of signs)
- I want to do it for all **m** examples, i.e. make an average at the end and minimize that
  - actually to make some math (later) easier, I will minimise 1/2 times this average

$$\text{minimize}_{\theta_0,\theta_1} \; \frac{1}{2}\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

m gives the size of the training set

Cost function

Let's write better what we need to minimise:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\downarrow$$

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

In other words, this is how the problem can be phrased:

- *"find me the values of* $\boldsymbol{\theta_0}$ *and* $\boldsymbol{\theta_1}$ *such that the function* **J** *is minimised"*

$$\text{minimize}_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

Basically, define a **cost function $J(\theta_0, \theta_1)$** that must be minimised

- this is a squared error cost function (you can think of other ones..)

x, θ_0, θ_1, h, J

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Important:
h is a function of x (given the θs),
J is a function of θs

θs seen in the x-y "space" (plane): a line. Each set of θs gives a different "fit" of the hypothesis to the training dataset in the x-y plane.

θs seen in the J space: a point. Each set of θs gives a single point in the J (θ_0, θ_1) space.

In this case, so far, "set of θ_s" = the couple (θ_0, θ_1)

A note

Why do we take the squares of the errors?

- it turns out that this squared error cost function is a reasonable choice and works well for most regression programs
- There are other cost functions that will work pretty well. But the square cost function is probably the most commonly used one for regression problems

Is it intuitive?

Check intuition.. how?

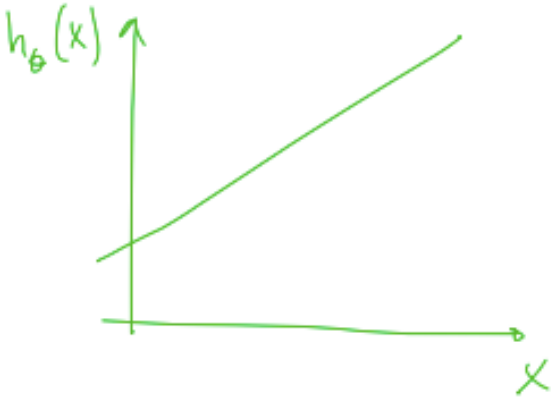
STANDARD

hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

parameters θ_0, θ_1

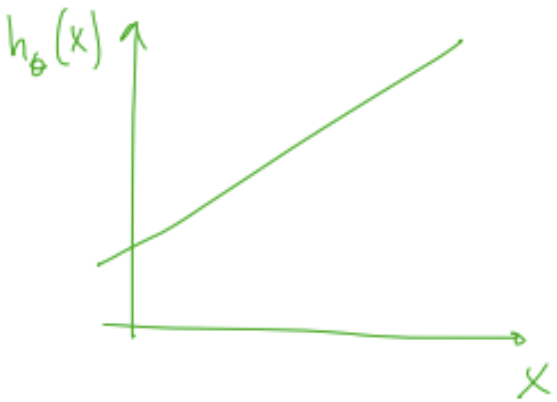
cost function $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

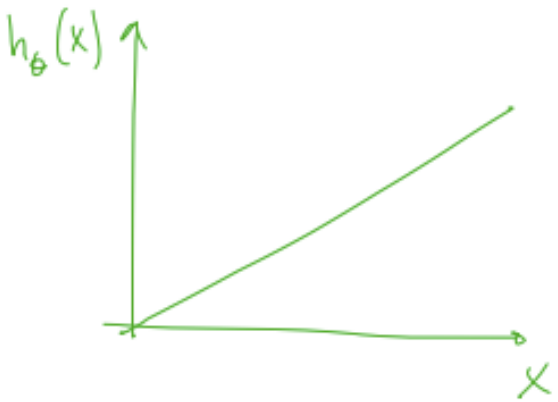
goal $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$



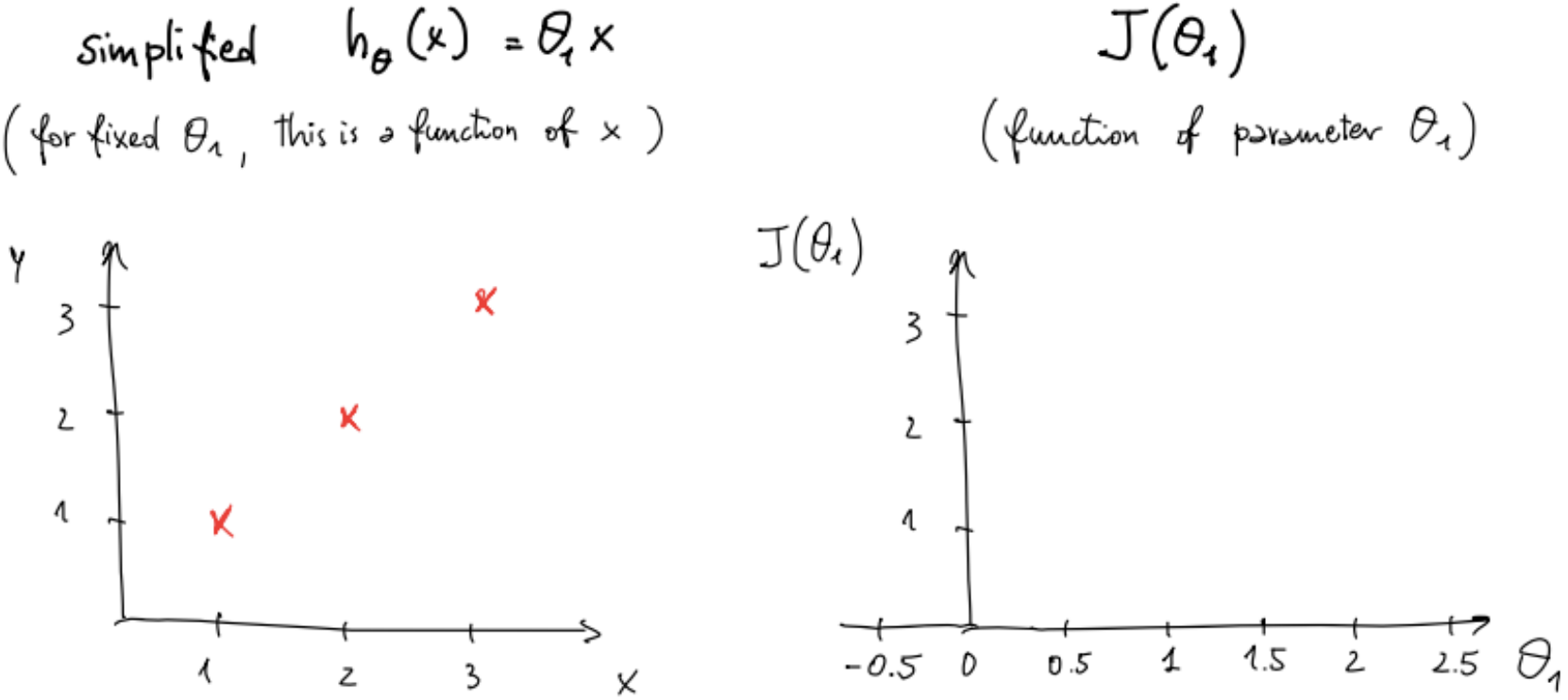
Check intuition.. with a simplified hypothesis function

	STANDARD	SIMPLIFIED
hypothesis	$h_\theta(x) = \theta_0 + \theta_1 x$	$h_\theta(x) = \theta_1 x$ ($\theta_0 = 0$)
parameters	θ_0, θ_1	θ_1
cost function	$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$	$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
goal	minimize $_{\theta_0, \theta_1} J(\theta_0, \theta_1)$	minimize $_{\theta_1} J(\theta_1)$





Let's keep this simplified hypothesis for a sec



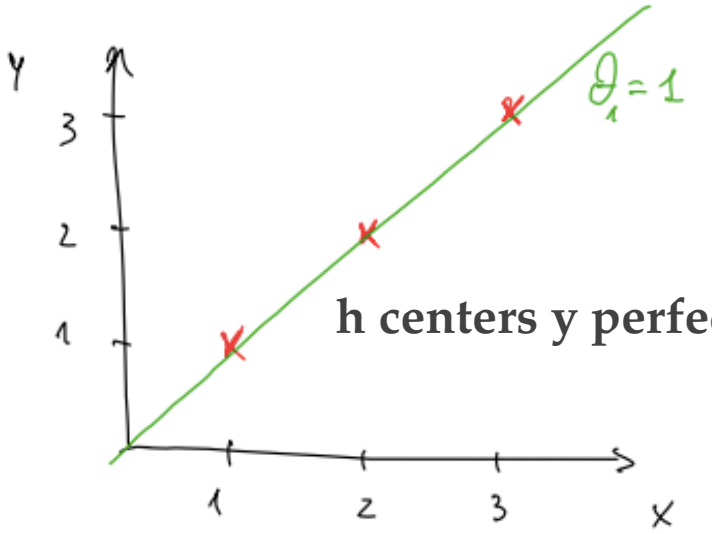
Use the simplified hypothesis for a while.
Assume m=3 examples in your training set - very trivial BTW (see left).
You want to see how J as a function of θ_1 looks like (see right)

Simplified $h_\theta(x) = \theta_1 x$

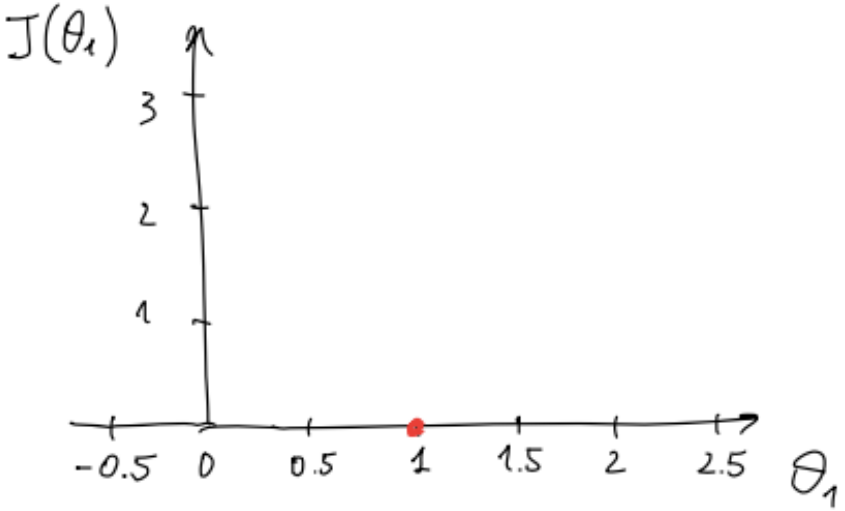
(for fixed θ_1, this is a function of x)

$J(\theta_1)$

(function of parameter θ_1)



h centers y perfectly!



$\boxed{\theta_1 = 1} \Rightarrow h_\theta(x) = x \Rightarrow y^{(i)} = h_\theta(x^{(i)}) = x^{(i)}$

$$J(\theta_1) = \frac{1}{2\times 3}\sum_{i=1}^{3}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2 = \frac{1}{6}\left(0^2 + 0^2 + 0^2\right) = 0$$

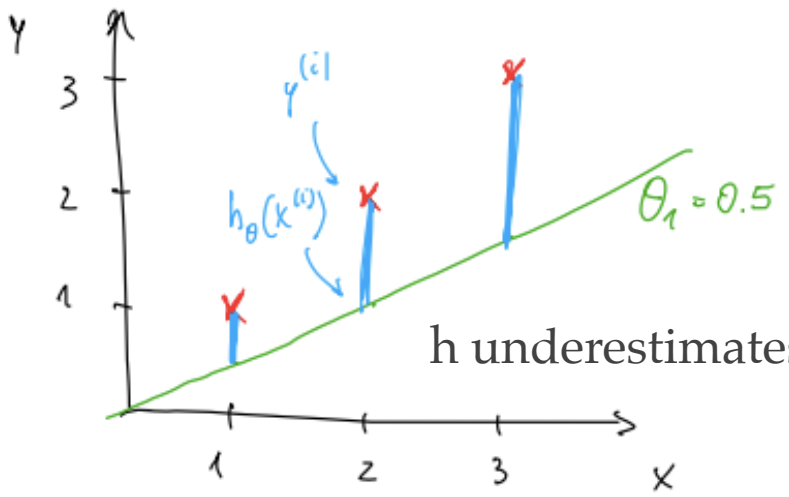
$\searrow \quad J(1) = 0$

simplified $h_\theta(x) = \theta_1 x$

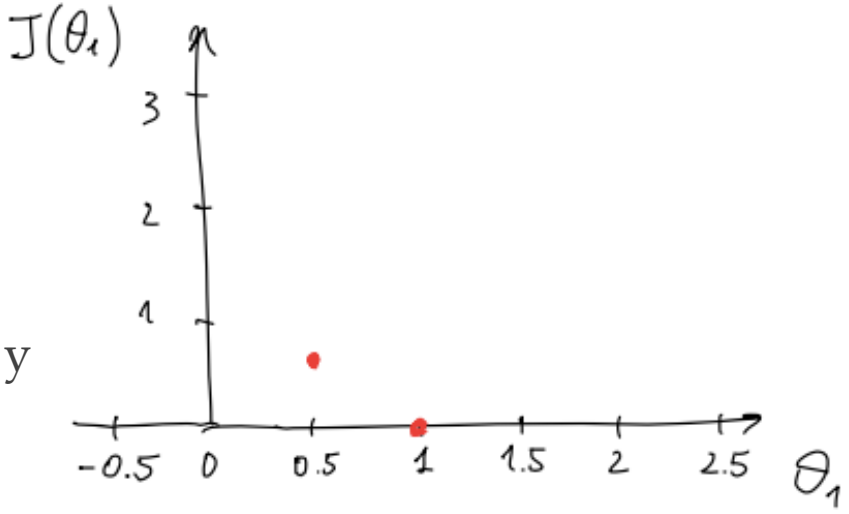
(for fixed θ_1, this is a function of x)

$J(\theta_1)$

(function of parameter θ_1)



h underestimates y



$\theta_1 = 0.5 \Rightarrow h_\theta(x) = 0.5x \Rightarrow y^{(i)} > h_\theta(x^{(i)}) = 0.5\,x^{(i)}$

$$J(\theta_1) = \frac{1}{2\times 3}\sum_{i=1}^{3}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2 = \frac{1}{6}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] = \frac{3.5}{6} \approx 0.58$$

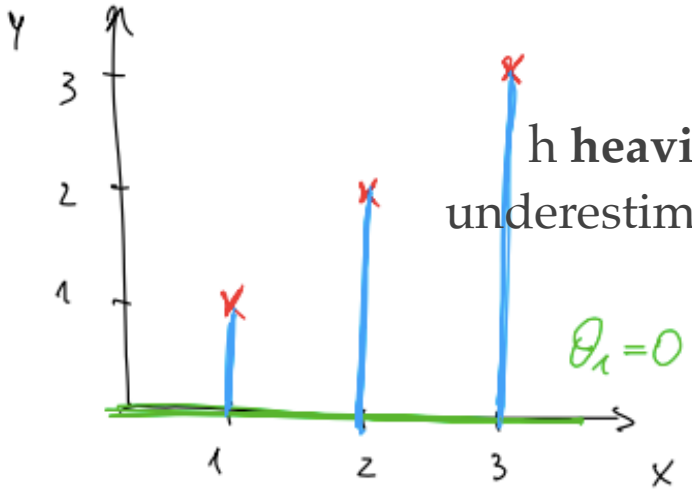
$J(0.5) \approx 0.58$

Simplified $h_\theta(x) = \theta_1 x$

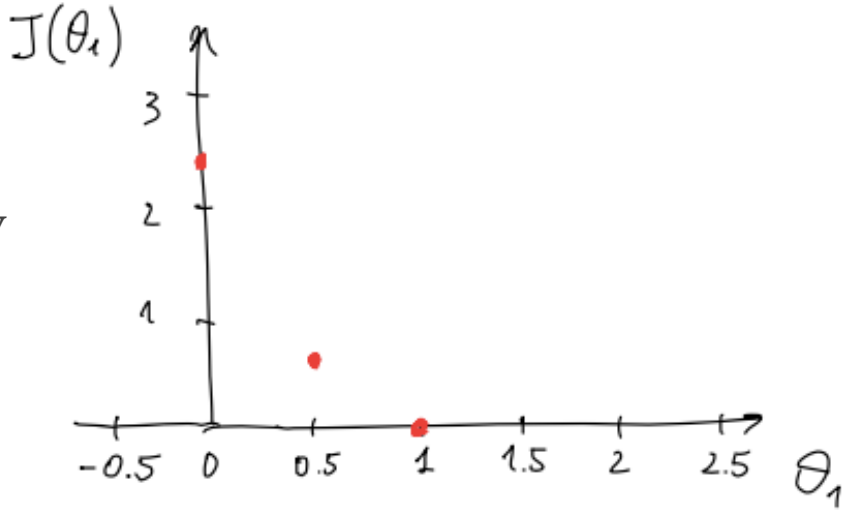
(for fixed θ_1, this is a function of x)

$J(\theta_1)$

(function of parameter θ_1)



h **heavily** underestimates y

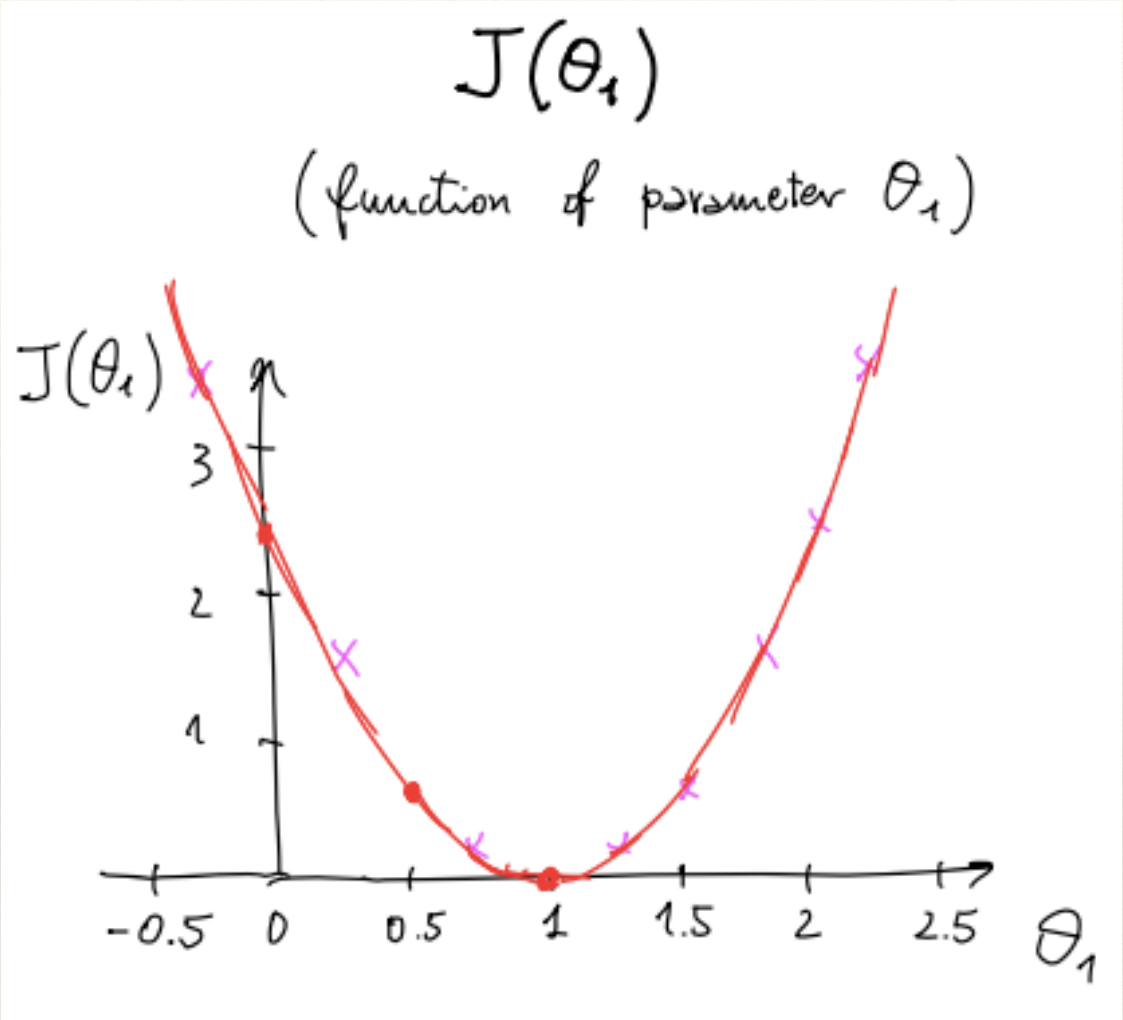


$\boxed{\theta_1 = 0} \Rightarrow h_\theta(x) = 0 \Rightarrow y^{(i)} > h_\theta(x^{(i)}) = 0$

$$J(\theta_1) = \frac{1}{2\times 3}\sum_{i=1}^{3}\left(\underbrace{\theta_1}_{0}x^{(i)} - y^{(i)}\right)^2 = \frac{1}{6}\left[1^2+2^2+3^2\right] \approx 2.3$$

$J(0) \approx 2.3$

Continue, and you get:



Each value of θ_1 gives a value of J, corresponding to a different hypothesis h

But we wanted to minimise!

That seems to happen for $\theta_1=1$, and indeed that corresponded to the "perfect fit".

Now that the basic thinking is clear, let's complicate things back a bit.

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

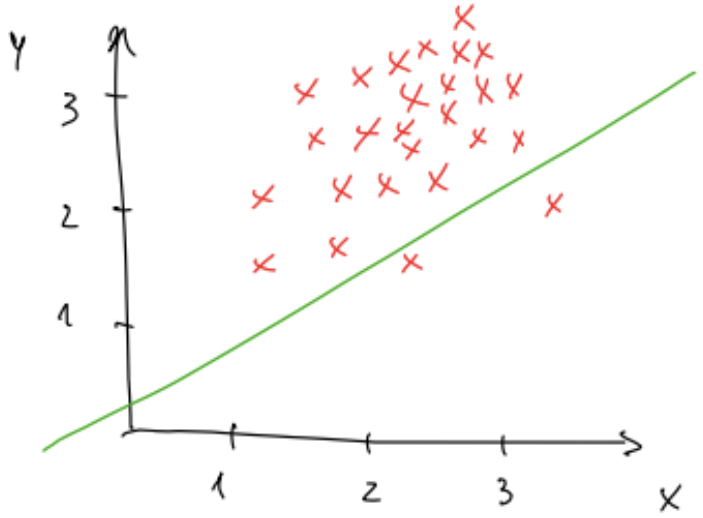
Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Now we want to keep **both parameters** θ_0 and θ_1 as we generate our intuition/visualisation for the cost function.

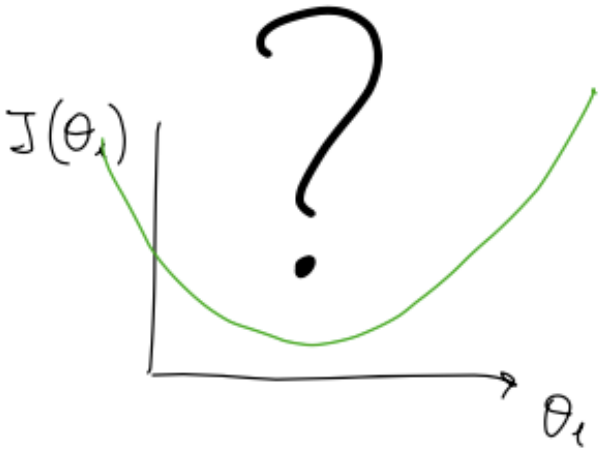
not simplified $h_\theta(x) = \theta_0 + \theta_1 x$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

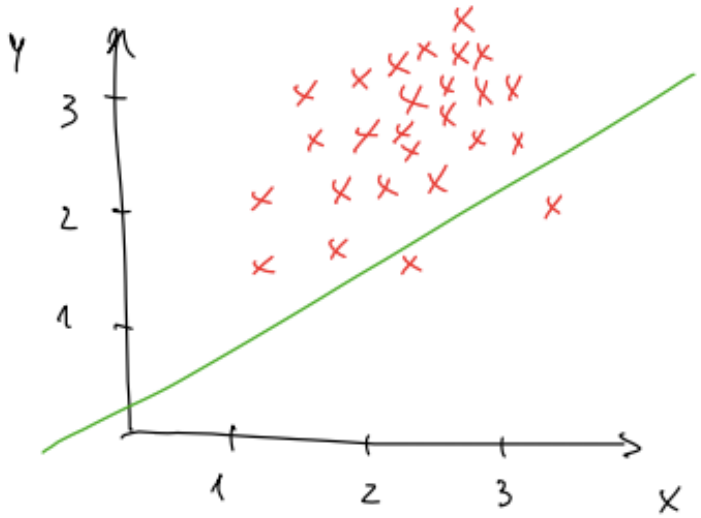
(function of parameter θ_0, θ_1)



It was easy before, as J was function of θ_1 only.. what about now?

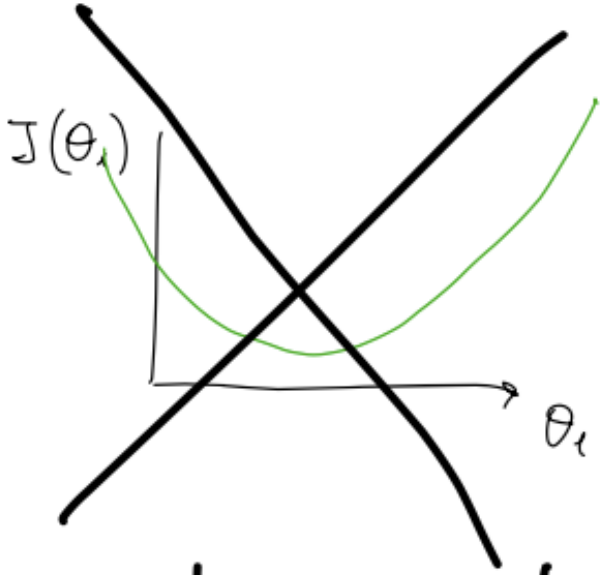
not simplified $h_\theta(x) = \theta_0 + \theta_1 x$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of parameter θ_0, θ_1)

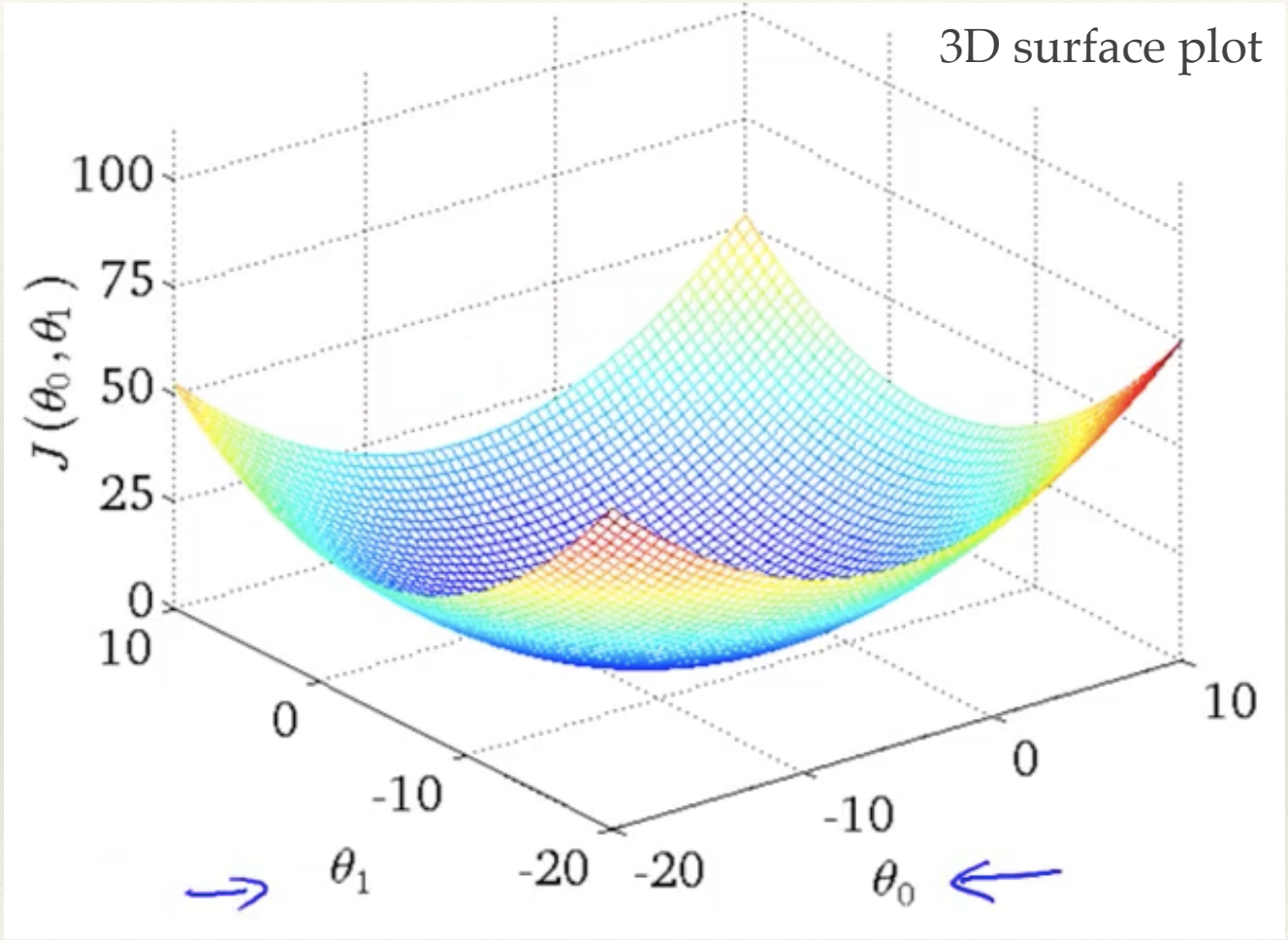


not so simple!

How can it look with θ_0, θ_1 ?

→

Perhaps more like this one:

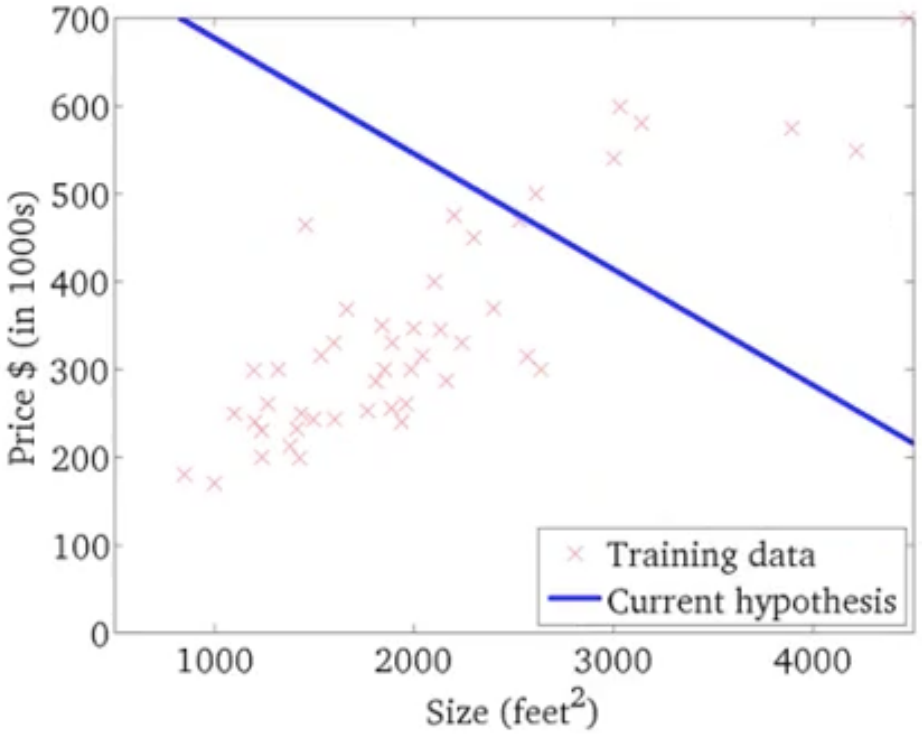


[ANG1]

Much easier for intuition to use visualisation such as "contour plots": see next

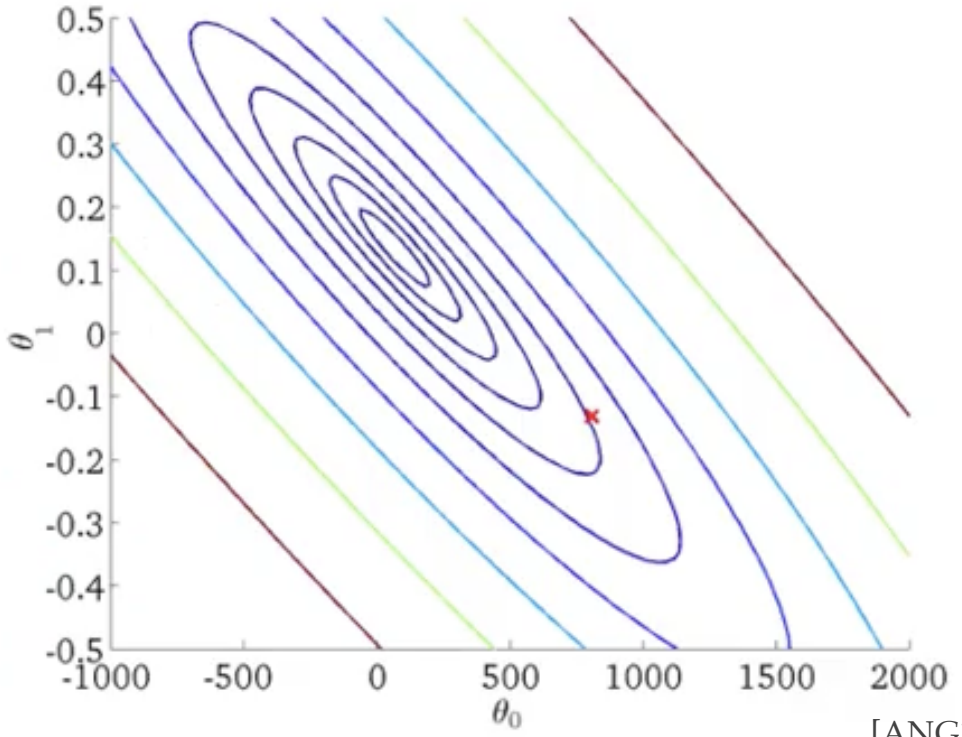
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

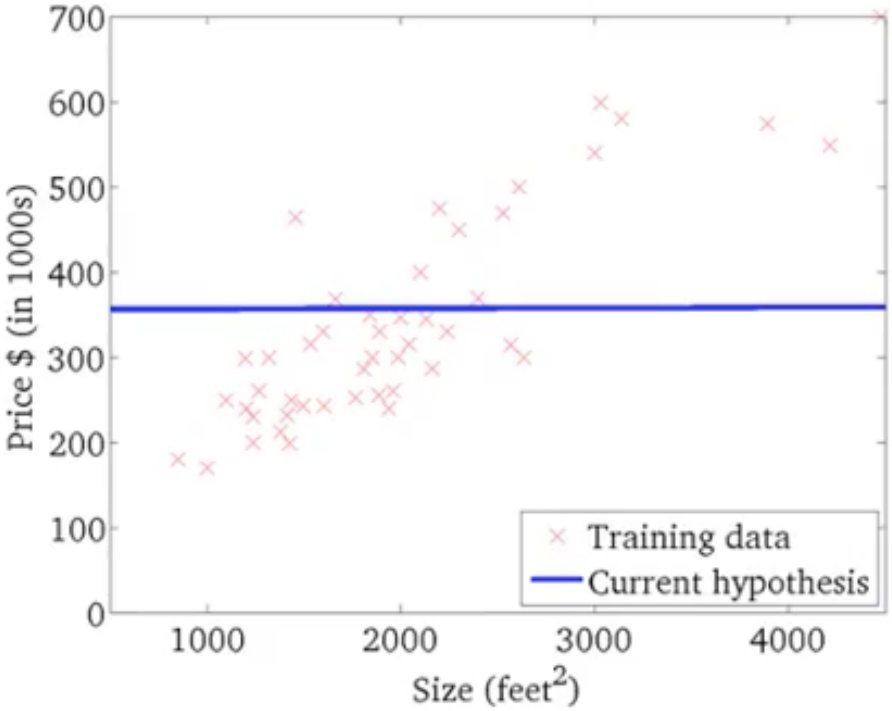


[ANG1]

Use contour plots for J (ovals are equi-cost (J)). Look at one point on an oval: it corresponds to θ_0 and θ_1 values that correspond to the blue **h** line on the left: not so good as a fit, and indeed the point in the J contour plot is pretty far from the minimum…

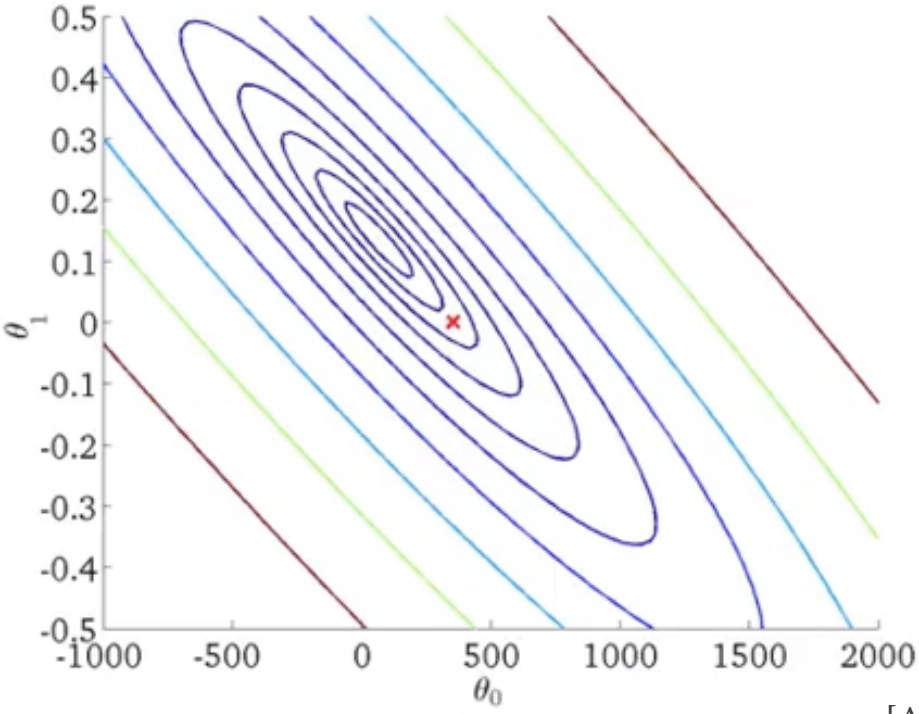
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

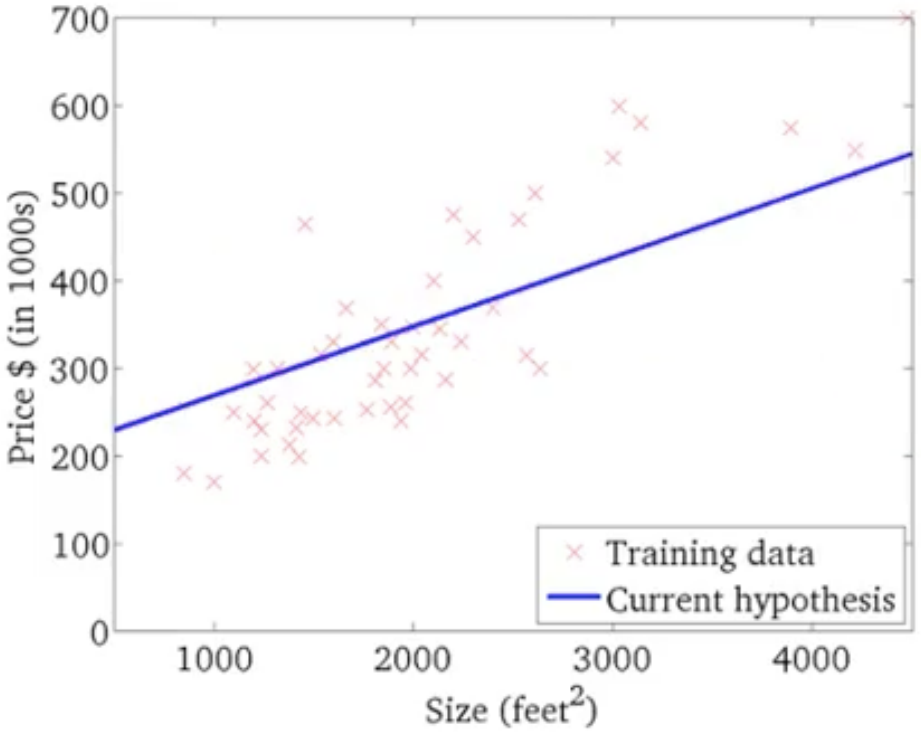


[ANG1]

Better. Note also we have $\theta_1 = 0$ here, so horizontal h(x)

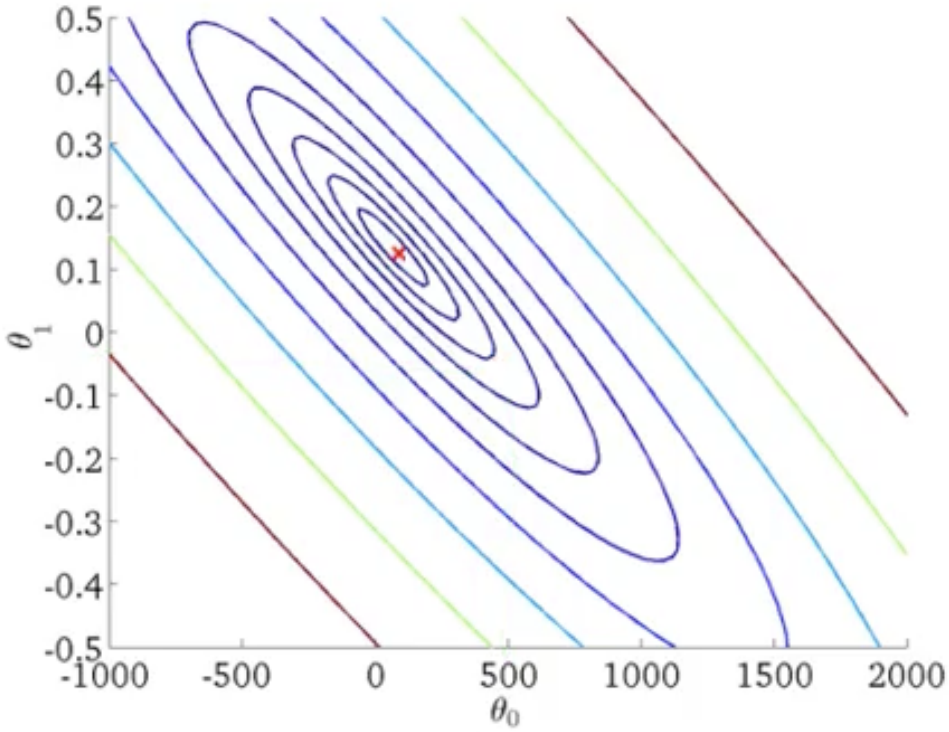
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



[ANG1]

MUCH better! Not yet at the minimum, actually, but pretty close..

Did you get the intuition?

If so..

NEXT: Gradient Descent.