# Introduction to Supervised Learning

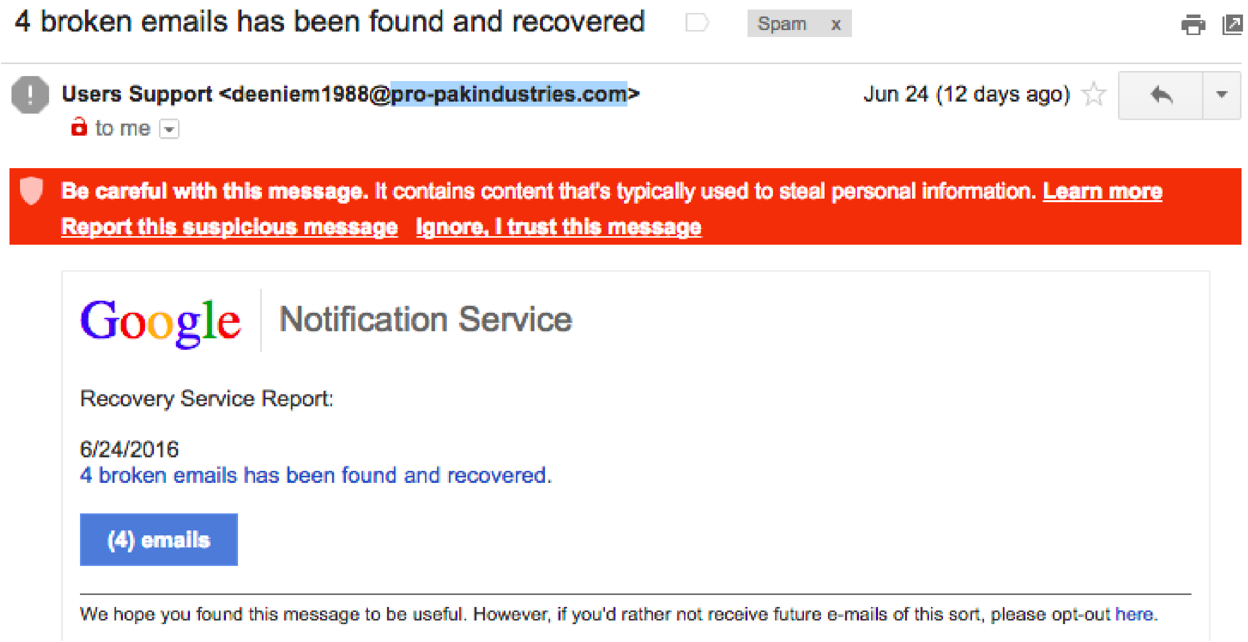
Ata Kaban

# What Is Supervised Learning?

- One of the most prevalent forms of ML
  - Teach a computer to do something, then let it use its knowledge to do it
  - Also called “learning with a teacher”
- Other forms of ML
  - Unsupervised learning (“learning without a teacher”)
  - Reinforcement learning (“learning with (delayed) feedback”)

# Example: Spam detection

- Input: Emails received




- Output: “Spam”, or “No spam”

# Example: Stock price prediction

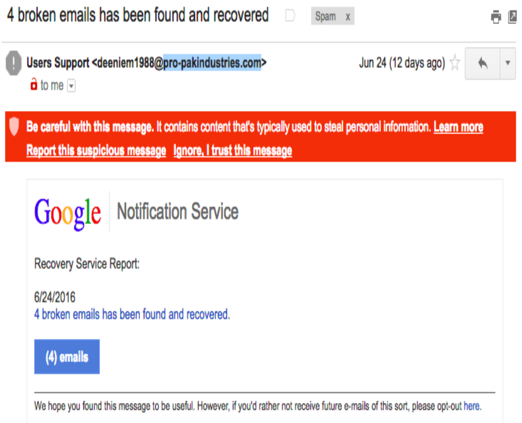
- Input: Historical records of stock prices



- Output: Next day's stock price

## Spam detection

- Input: Emails received




- Output: “Spam”, or “No spam”
- This is a **classification** problem. The output has 2 possible values

## Stock price prediction

- Input: Historical records of stock prices


- Output: Next day’s stock price
- This is a **regression** problem. The output is a real value.

# Types of supervised learning

- Regression
- Classification
  - Binary
  - Multi-class
  - …

# Supervised learning

Task:

- Given some **input** *x*,
- Predict an appropriate **output** *y.*

*Want: a **function** f such that f(x)=y*

*Have: examples of input-output pairs* $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$

*Supervised learning helps find a good f.*

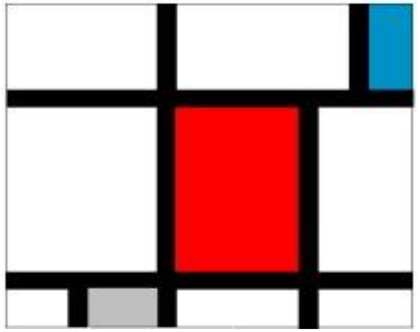
# Training data

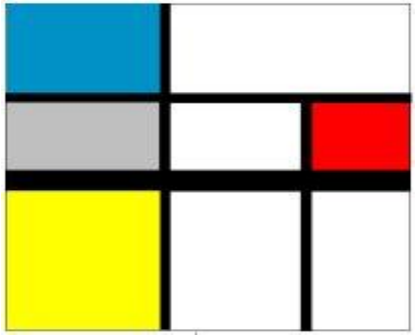
- Supervised learning needs annotated data for training:

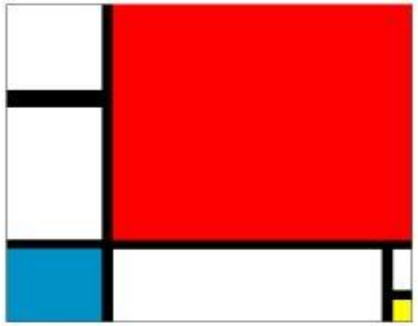
  in the form of examples of (Input, Output) pairs

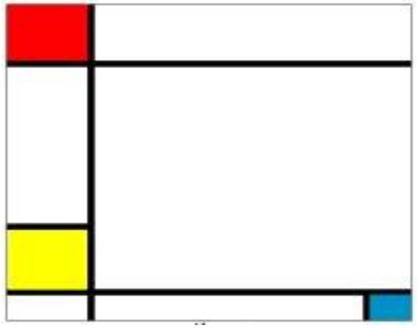
- After training completed,
  - you present it with new Input that it hasn't seen before
  - It needs to predict the appropriate Output

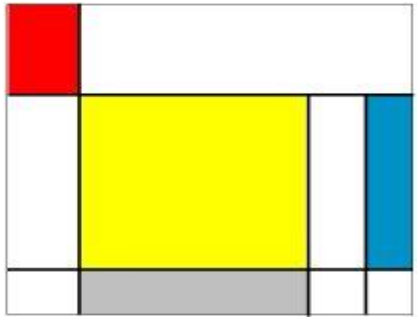
# Is painting 8 a genuine Mondrian?

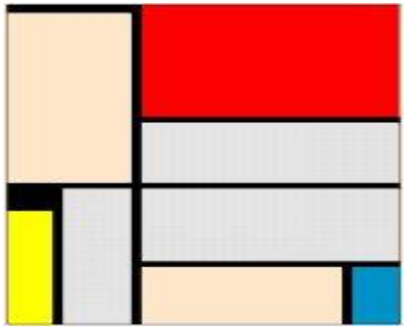

1. No

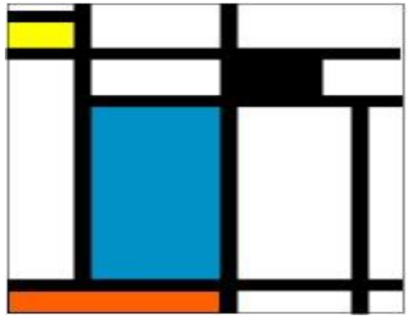

2. No

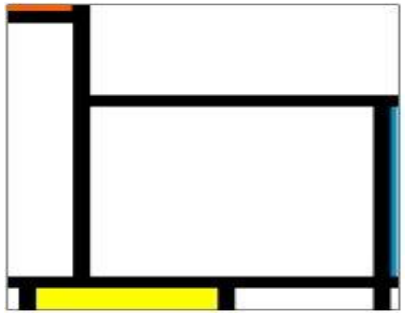

3.Yes


4. Yes


5.No


6. Yes


7. No


8. ?

Attributes

Labels

Annotated training data

Examples

| Number | Lines | Line types | Rectangles | Colours | Mondrian? |
|---|---|---|---|---|---|
| 1 | 6 | 1 | 10 | 4 | *No* |
| *2* | 4 | 2 | 8 | 5 | *No* |
| 3 | 5 | 2 | 7 | 4 | Yes |
| 4 | 5 | 1 | 8 | 4 | Yes |
| *5* | 5 | 1 | 10 | 5 | *No* |
| 6 | 6 | 1 | 8 | 6 | Yes |
| *7* | 7 | 1 | 14 | 5 | *No* |

Painting 8

| Number | Lines | Line types | Rectangles | Colours | Mondrian? |
|---|---|---|---|---|---|
| 8 | 7 | 2 | 9 | 4 | **???** |

# How quick will your team complete a project?

| (programming language) | (team expertise) | (estimated size) | … | (required effort) |
|---|---|---|---|---|
| Java | low | 1000 | … | 10 p-month |
| C++ | medium | 2000 | … | 20 p-month |
| Java | high | 2000 | … | 8 p-month |
| … | … | … | … | … |

# General notation we will use

| (programming language) | (team expertise) | (estimated size) | … | (required effort) |
|---|---|---|---|---|
| | | $\boldsymbol{x}^{(1)}$ | | $y^{(1)}$ |
| | | $\boldsymbol{x}^{(2)}$ | | $y^{(2)}$ |
| | | $\boldsymbol{x}^{(3)}$ | | $y^{(3)}$ |
| … | … | … | … | … |

Vector notation

$$\boldsymbol{x}^{(i)} = \Big(\underbrace{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_d^{(i)}}_{\text{Attributes}}\Big)$$ The input of the *i*-th example

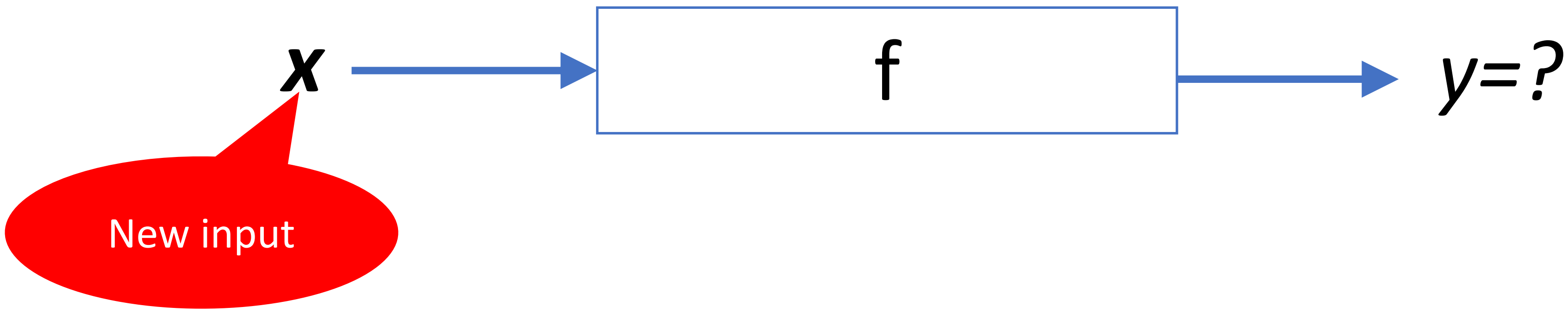
# Workflow of supervised learning:
# 1. Training phase

$$\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$$

$$\Big\downarrow \text{ ML Algorithm}$$

$$\boxed{\quad f \quad}$$

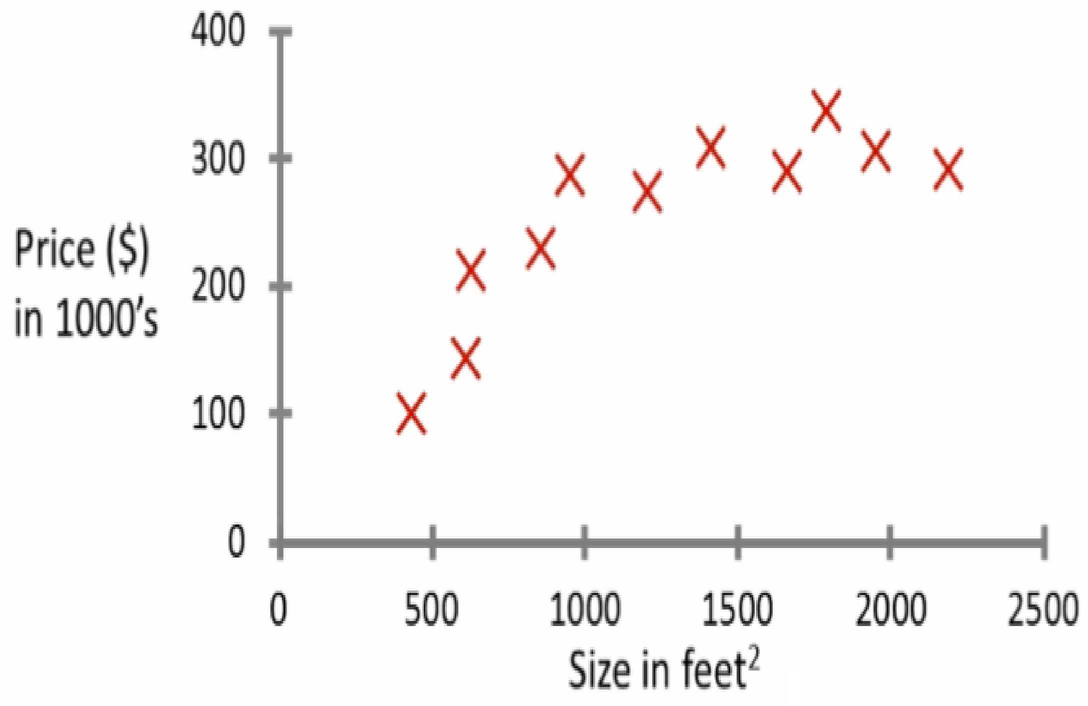
# Workflow of supervised learning:
# 2. Test phase & use



# Pictorially

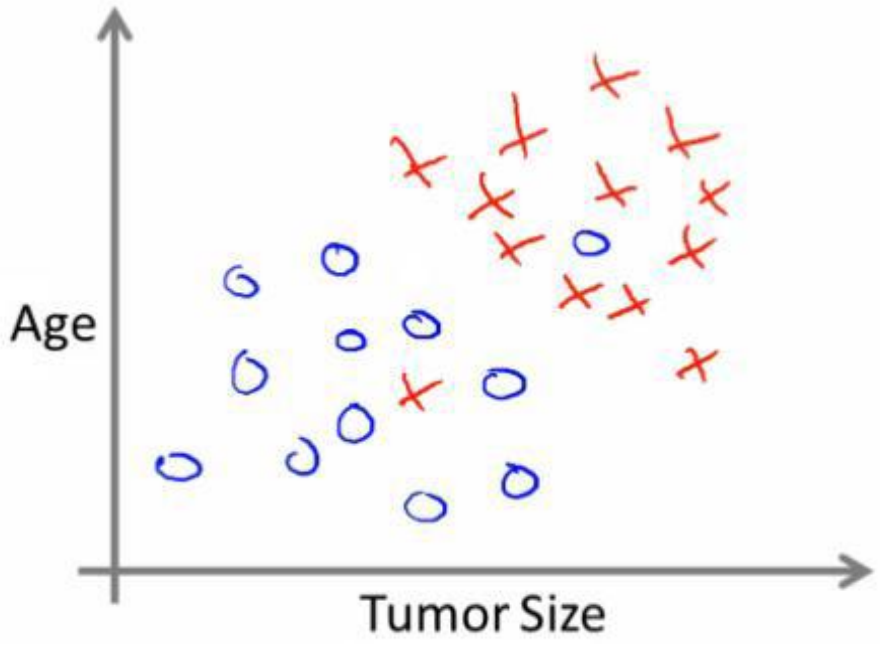
- Regression problem

Housing price prediction.



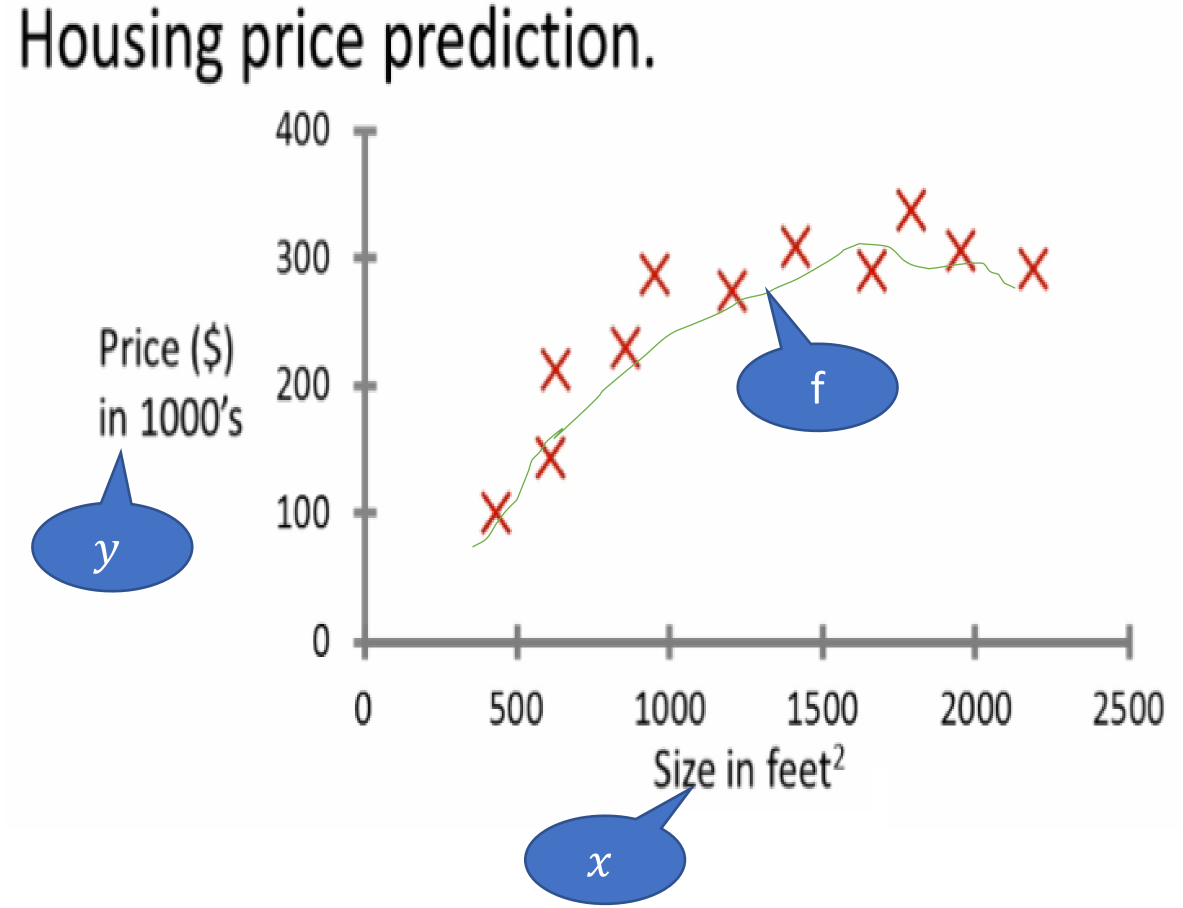

- Classification problem

Breast cancer prediction



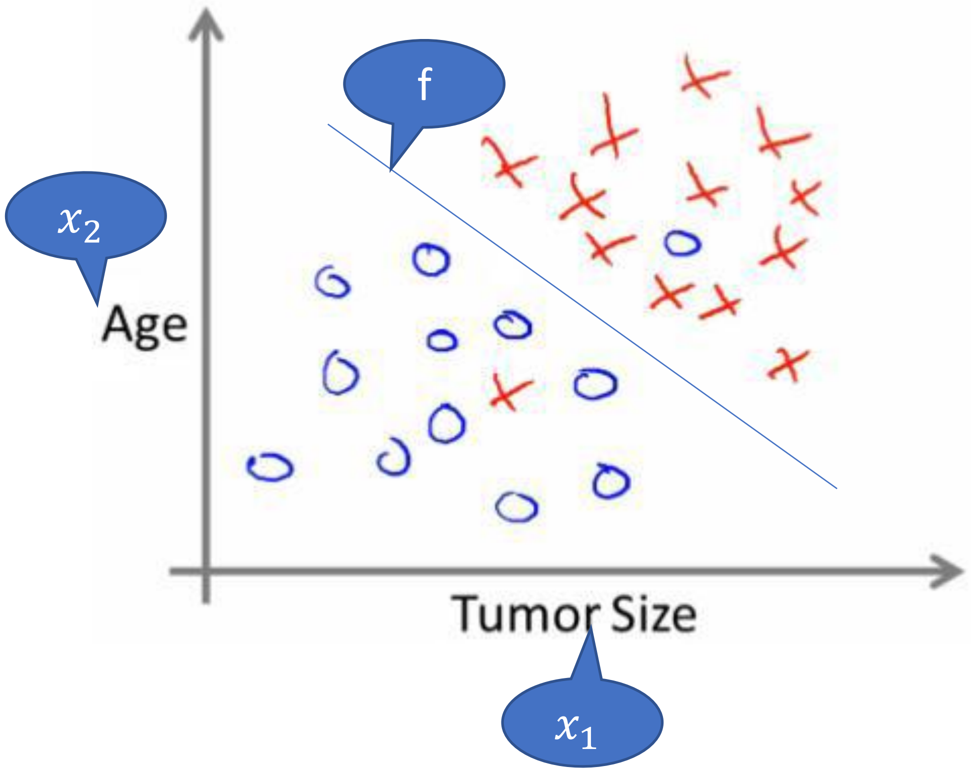
# Pictorially

- Regression problem




- Classification problem

Breast cancer prediction



# Terminology in Supervised Learning

- Input = attribute(s) = feature(s) = independent variable
- Output = target = response = dependent variable
- function = hypothesis = predictor

# Pause. Is this some magic?

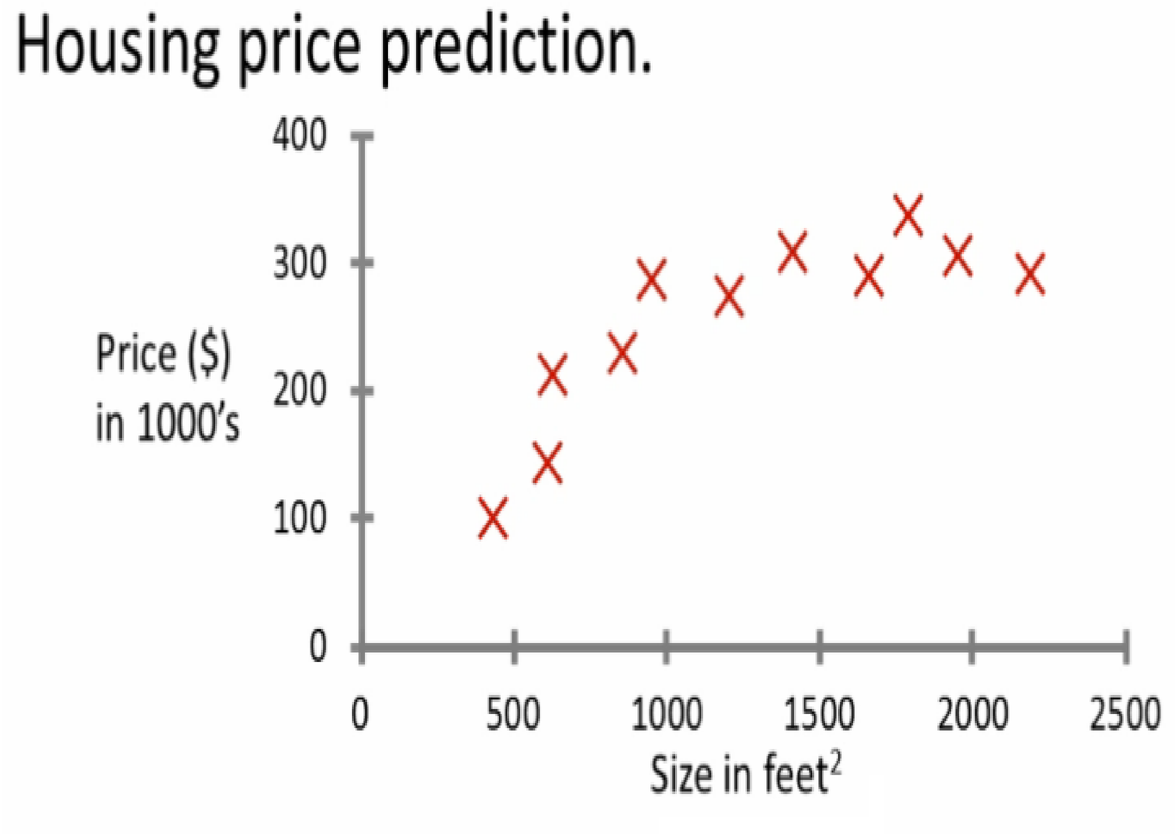
So…

- there is this unknown function we’re after
- we are given the function values at *n* specific points only (training set)
- is it really possible to find out the function values at other points?

- No!
- Not unless we make the right assumptions about the unknown function
- Each ML algorithm, implicitly or explicitly, makes assumptions.
- There is a zoo of ML algorithms, there is no best ML algorithm
- Our goal is to focus on few of them, and understand how they work

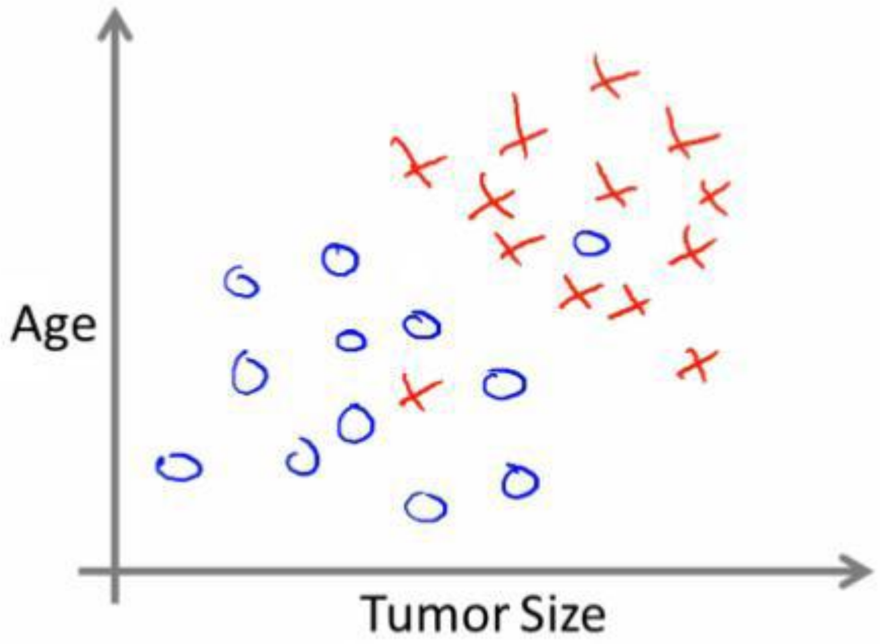
# How many predictors are there for these data?

- Regression problem

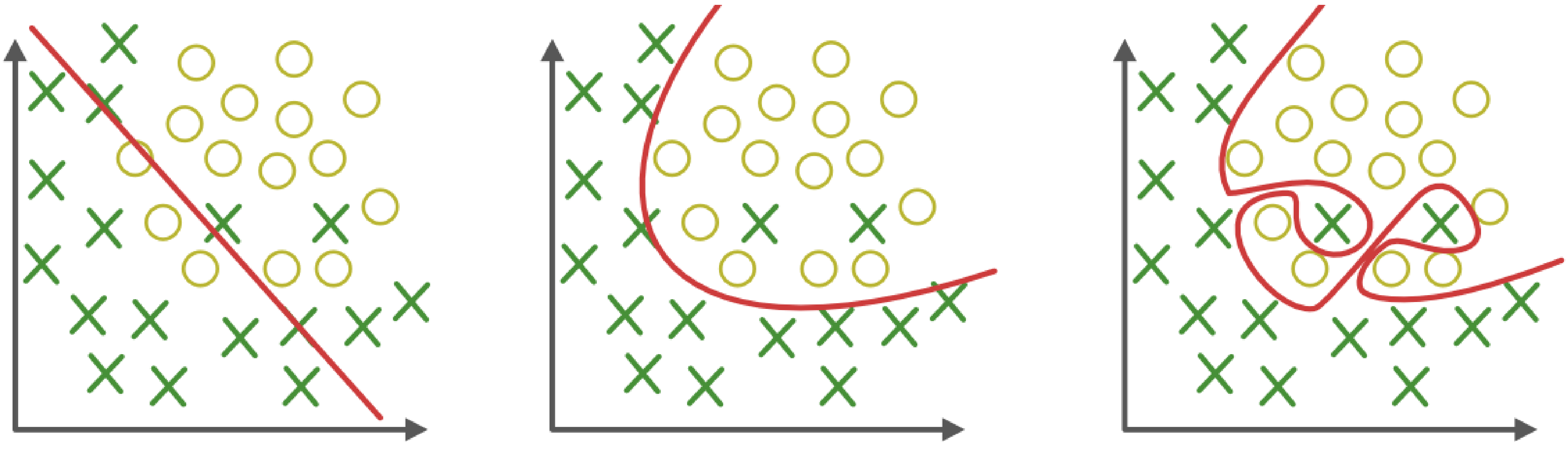



- Classification problem

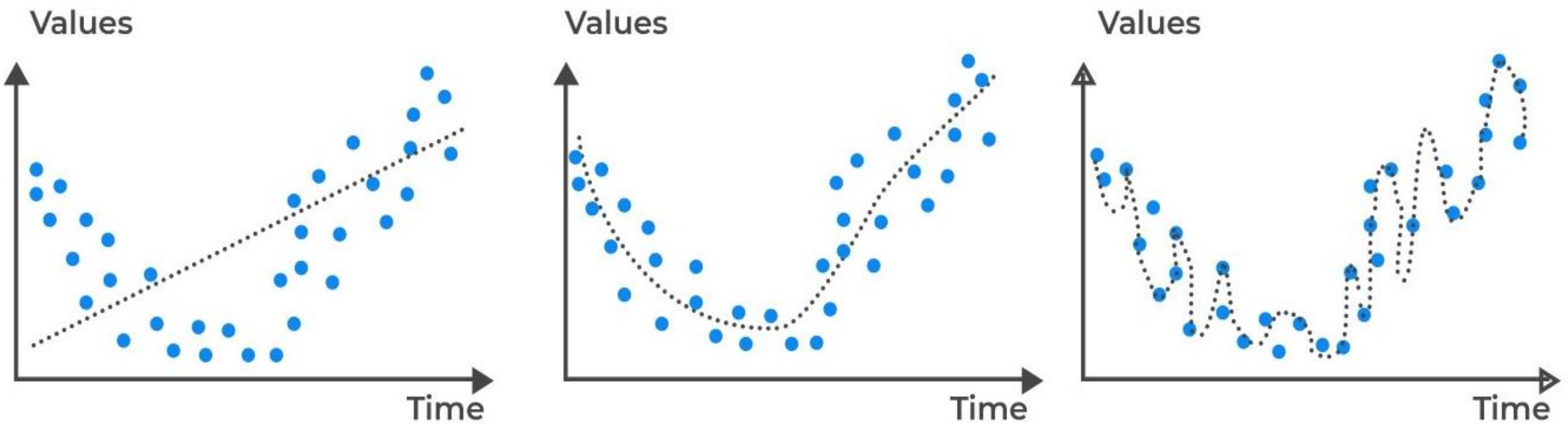
Breast cancer prediction



Classification



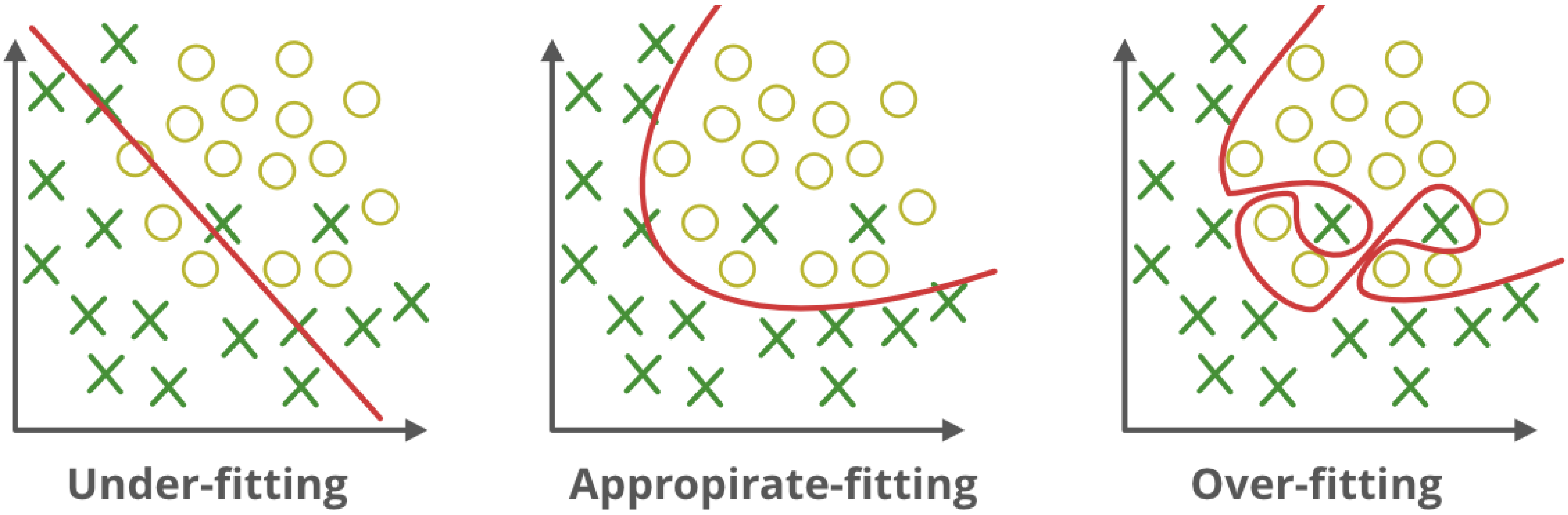
Regression



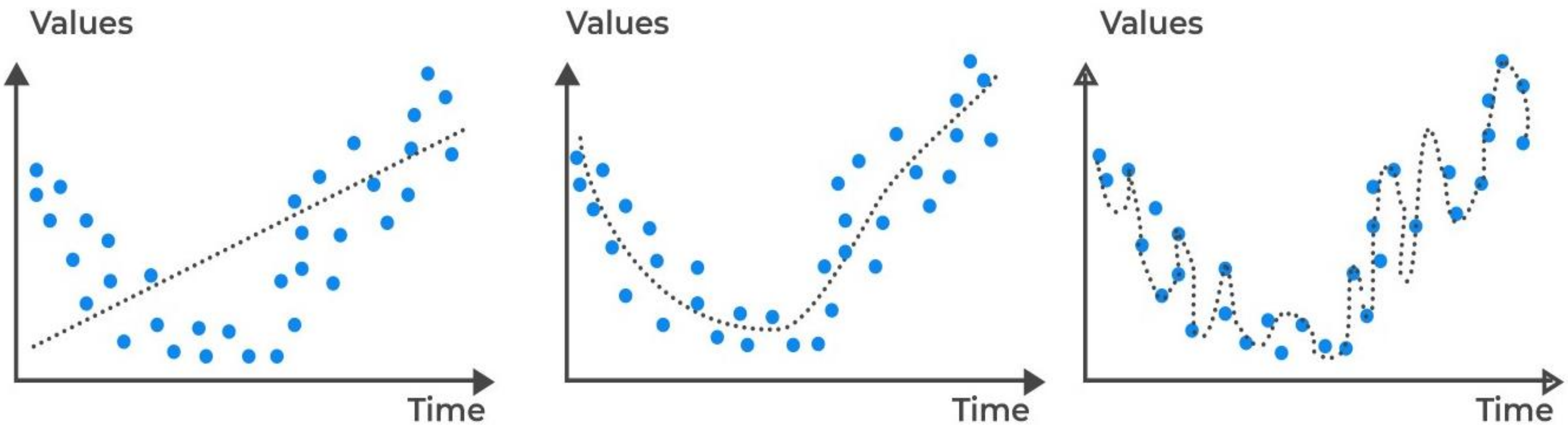

https://www.geeksforgeeks.org/regularization-in-machine-learning/

https://analystprep.com/study-notes/wp-content/uploads/2021/03/Img_13.jpg

Classification




Regression




https://www.geeksforgeeks.org/regularization-in-machine-learning/

https://analystprep.com/study-notes/wp-content/uploads/2021/03/Img_13.jpg

# Overfitting and underfitting

- Fitting the training data too well is BAD! Why?
- Remember the data you actually want to classify, or predict for, is not the same as the training data – so learning every irrelevant detail (noise) in a training data set will not help
- Overfitting happens when the model is more complex than required
- Underfitting happens when the model is simpler than required

# Applications of supervised learning

- Handwriting recognition
  - When you write an envelope, algorithms can automatically route envelopes through the post
- Computer vision & graphics
  - When you go out during lockdown, object detection & visual tracking algorithms can automatically detect compliance with the rules
- Bioinformatics
  - Algorithms can predict protein function from sequence
- Human-computer interaction
  - Intrusion detection algorithms can recognise speech, gestures, intention

# Prevalence of ML

- Generality
  - E.g. a robot learning to navigate mazes must be able to learn the layout of the maze it encounters
- Adaptability
  - E.g. a program designed to predict tomorrow's stock market must learn to adapt when conditions change from boom to bust
- Applicability
  - Often the human programmer has no idea how to program a solution to the problem (think of how you recognise your friend's face)