

The background of the slide features a dark, textured surface. On the left, several teal lines with a stepped, circuit-like pattern flow horizontally. In the center, a dense network of small teal nodes is interconnected by thin lines, with a few nodes highlighted in orange. To the right, a single orange line curves upwards, while several teal lines flow horizontally, some appearing as wavy bands.

# Introduction to Machine Learning

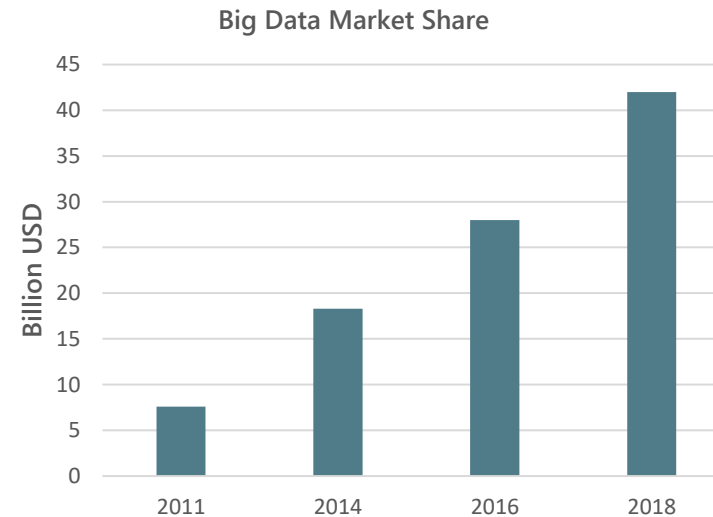
Özgür Martin

# Why Now?



# Big Data

- Data is learning material for AI algorithms.
- More data means more learning material for AI models.
- Big data thus enables more consistent and accurate AI models.





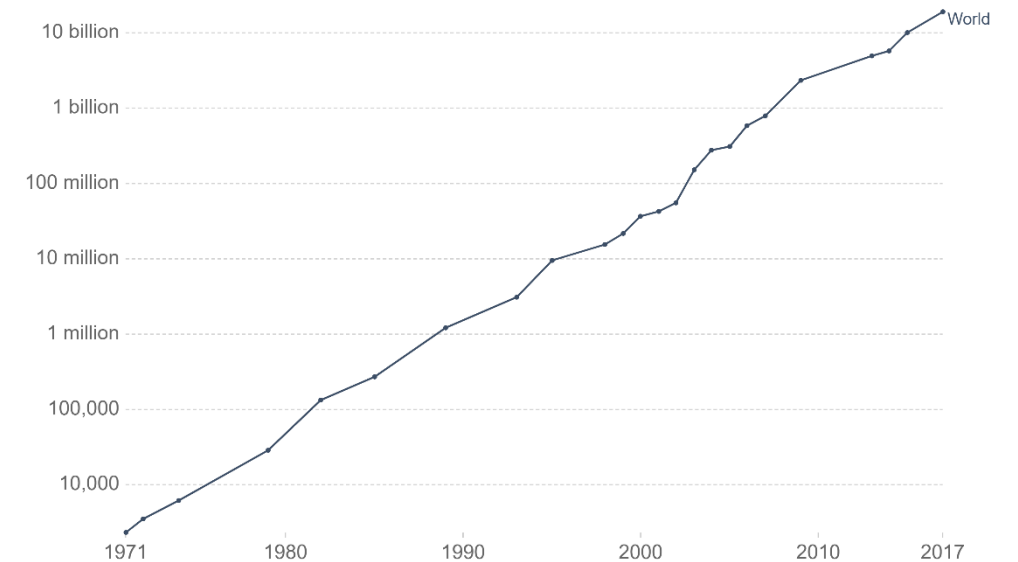
# Processing Power

Despite speculations, processing power has been steadily increasing exponentially.

## Moore's Law: Transistors per microprocessor

Number of transistors which fit into a microprocessor. This relationship was famously related to Moore's Law, which was the observation that the number of transistors in a dense integrated circuit doubles approximately every two years.

Our World  
in Data



Source: Karl Rupp. 40 Years of Microprocessor Trend Data.

CC BY

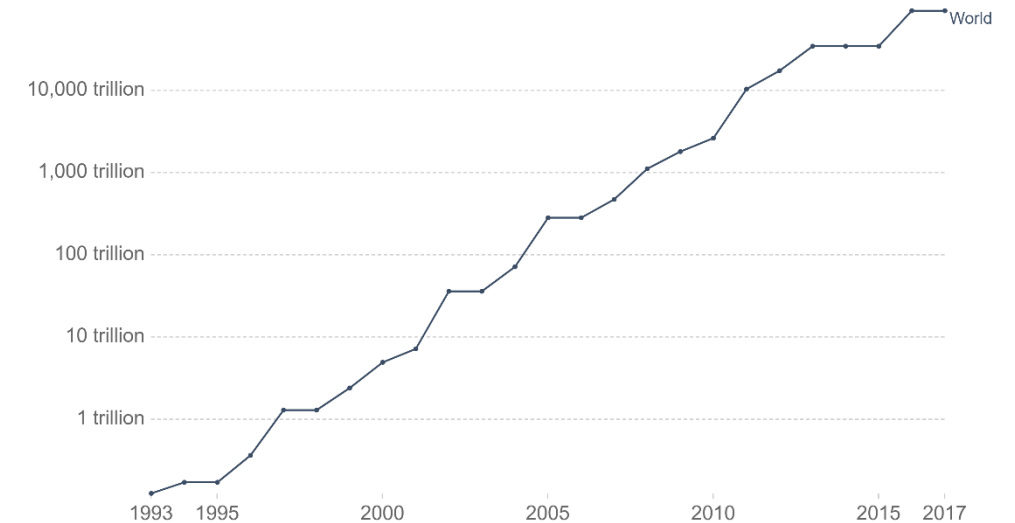
# Processing Power

Despite speculations, processing power has been steadily increasing exponentially.

## Supercomputer Power (FLOPS), 1993 to 2017

The growth of supercomputer power, measured as the number of floating-point operations carried out per second (FLOPS) by the largest supercomputer in any given year. (FLOPS) is a measure of calculations per second for floating-point operations. Floating-point operations are needed for very large or very small real numbers, or computations that require a large dynamic range. It is therefore a more accurate measure than simply instructions per second.

Our World  
in Data



Source: TOP500 Supercomputer Database

CC BY

# Theoretical Paradigm Shift

In the earlier days of AI, one needed a deep understanding of the problem in mathematical terms in order to reach useful conclusions using AI models.

---

## Deep Blue vs Kasparov - 1997



Deep Blue (and its successors) ran on an algorithm which uses a precise measure of «how good» a given position is in a game of chess.

Once you can measure how good positions are then it is not very difficult to select moves which lead toward better positions.

# Theoretical Paradigm Shift

Modern AI algorithms rely on «[statistical learning](#)» which, at least in certain instances, can overcome the lack of mathematical understanding of a problem.

---

Go



Go has long been known to be notoriously difficult for mathematical analysis.

In 1970's AI researchers thought computers could not possibly beat the best human players.

# Theoretical Paradigm Shift

Statistical learning, expressed through a specific class of models «deep neural networks», has proven to be extremely powerful by defeating Lee Sedol decisively.

---

## AlphaGo vs Lee Sedol - 2017



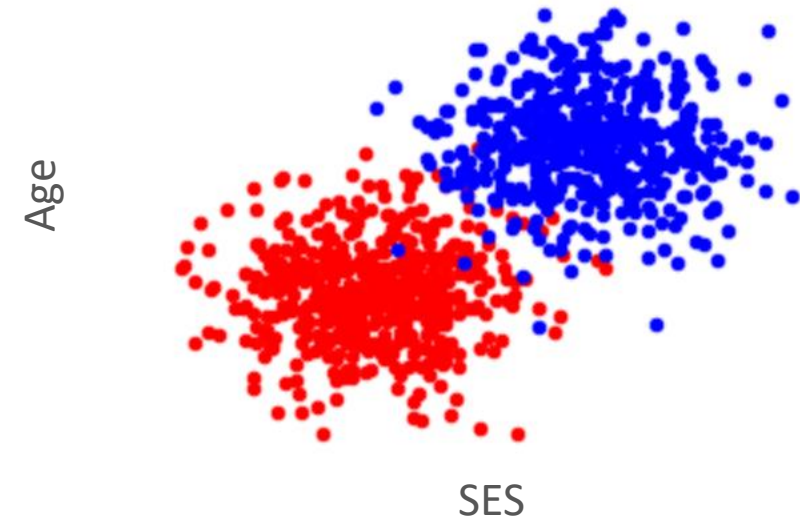
Lee Sedol, acknowledged that winning one game against AlphaGo was a big accomplishment.

It is now deemed impossible for humans to win a matchup against AlphaGo.



# Typical Statistical Learning Setup \*

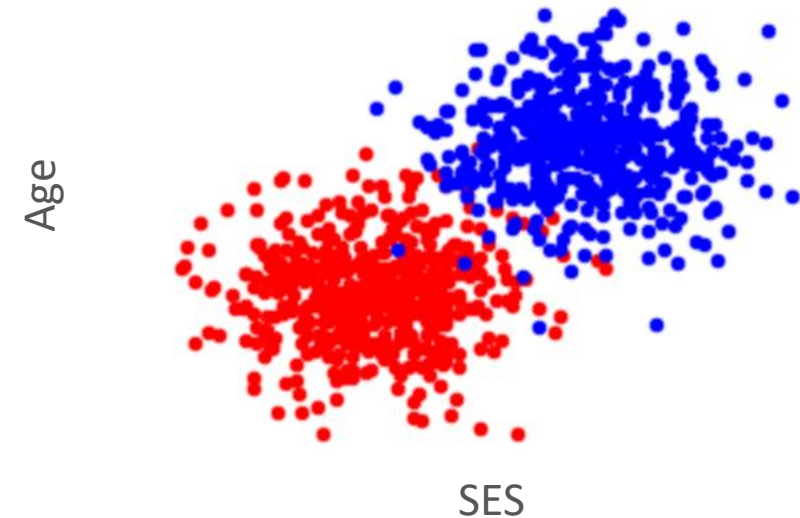
- We have two measurements on our subjects, say socioeconomic status and age. These are called **features**.
- We also observe a certain outcome variable for each subject, say using a certain product. This is called the **target variable**.
- The set of these observations (features and target) is called the **learning data**.



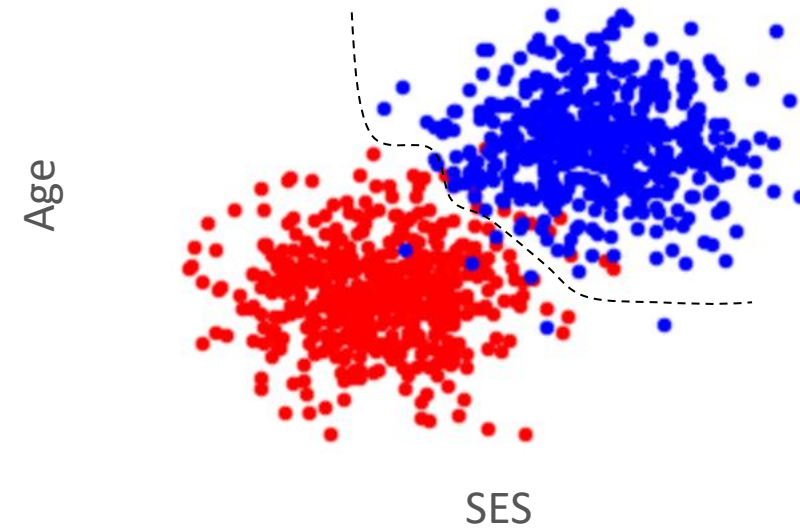
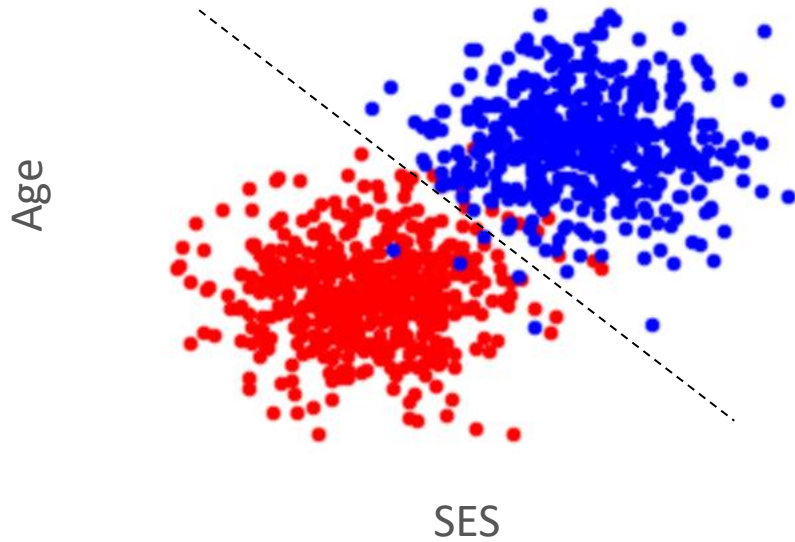
\* Also called **Supervised Learning**

# Typical Statistical Learning Setup

- An algorithm which gives a procedure to distinguish blue and red points looking at SES and Age variables is called an AI model.
- Such an algorithm «learns» from our observed data and then we use it to make predictions about new cases.

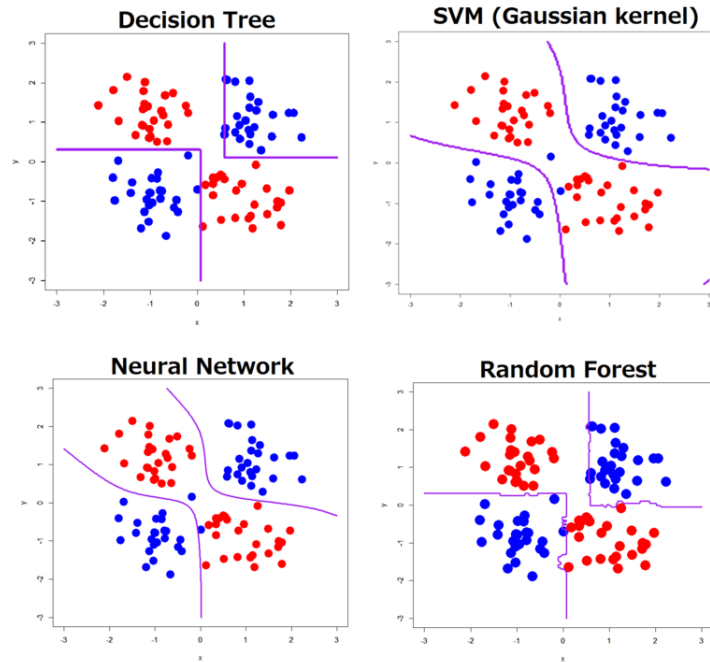


# Issue 1: Too Many Options



# Solve Issue 1: Restrict

In order to be able to find a solution our algorithms need to be restricted to a certain class of possible solutions.

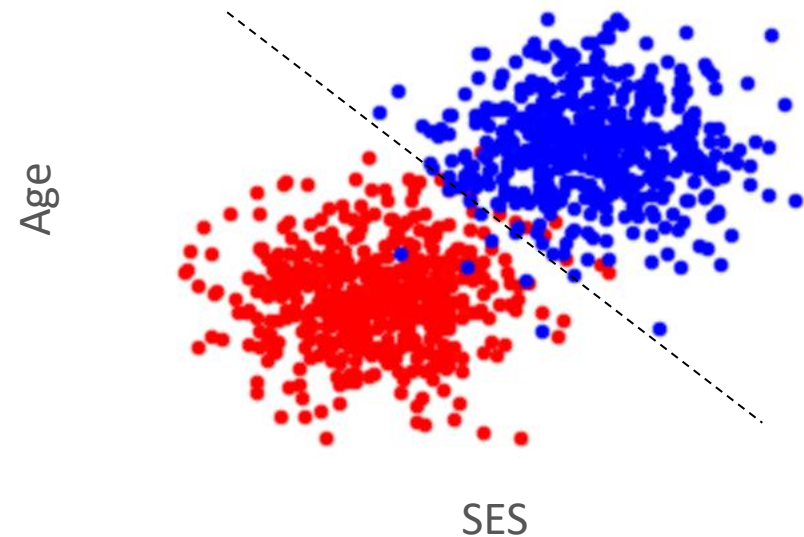


# Issue 2: The Best Solution

After we decide on the method (random forests and neural networks are the primary choices in modern applications) we need to find an ***optimal*** solution.

For example, if we decide to solve this problem with a line we want to find the line which gives minimum classification error.

***Gradient boosting*** and ***gradient descent*** are examples of optimization techniques used in modern AI applications.





# Issue 3: Data Availability

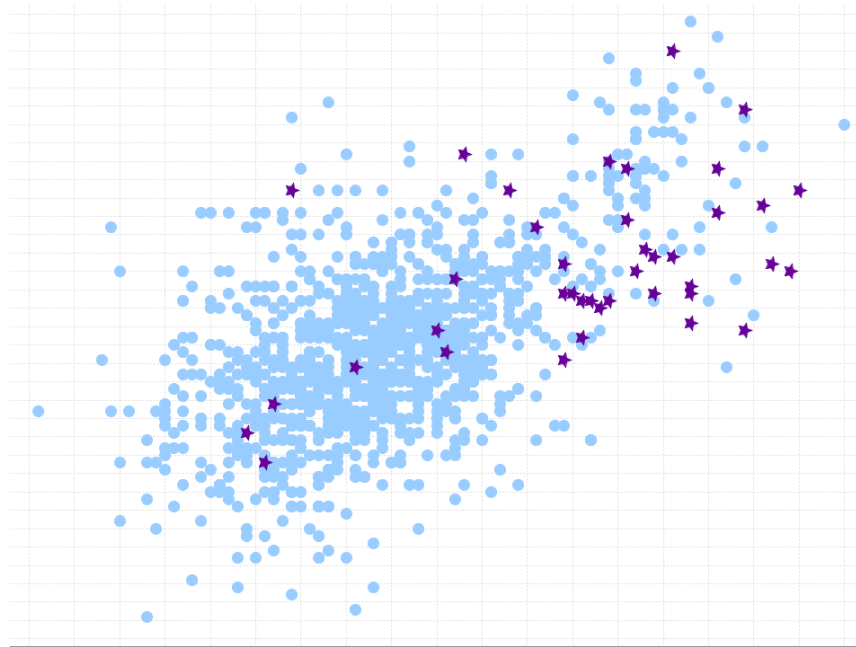
«If we have data let's look at data. If all we have are opinions, let's go with mine.»

*Jim Barksdale*

Statistical learning is based on the assumption that we do have actual data collected from real samples. No data often means no AI.

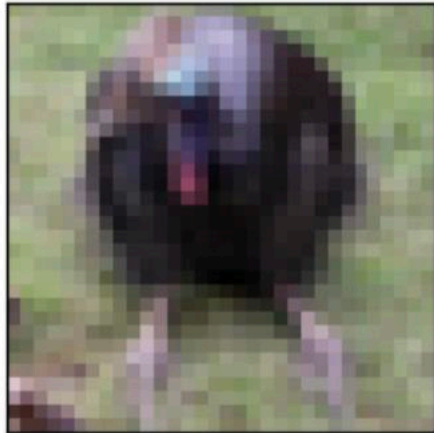
# Issue 4: The Problem Itself

We simply can not expect to obtain a reliable solution to every problem by collecting data.

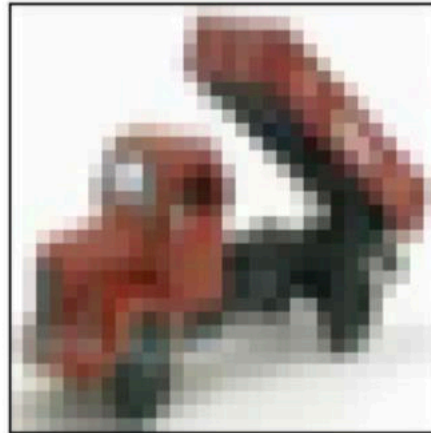


# A Machine Learning Example

The CIFAR-10\* dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class.



bird (2)



truck (9)



frog (6)

Image from [www.tensorflow.org](http://www.tensorflow.org)

\*Alex Krizhevsky, Learning multiple layers of features from tiny images, Tech. report, 2009.

# A Machine Learning Example

Denote the images in the dataset by

$$\{x_i\}_{i=1}^n \text{ with } x_i = (\underbrace{x_{i,1}, \dots, x_{i,d_x}}_{\text{pixels in the image}}) \in \mathbb{R}^{d_x}$$

Denote the labels by

$$\{y_i\}_{i=1}^n \text{ with } y_i \in \{1, \dots, 10\} \subset \mathbb{R}$$

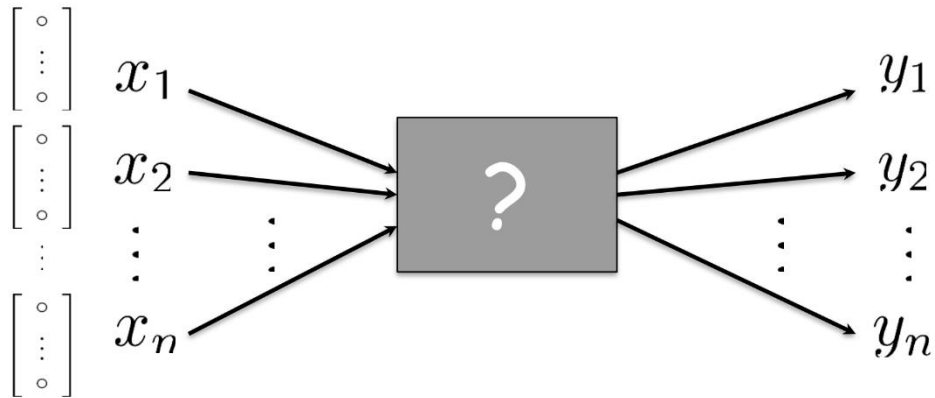
We call the set  $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^{d_x} \times \mathbb{R}$  as the **training points**.

$n=60000$  and  $d_x = 32 \times 32 \times 3 = 3072$  since every pixel has 3 values for RGB colors.

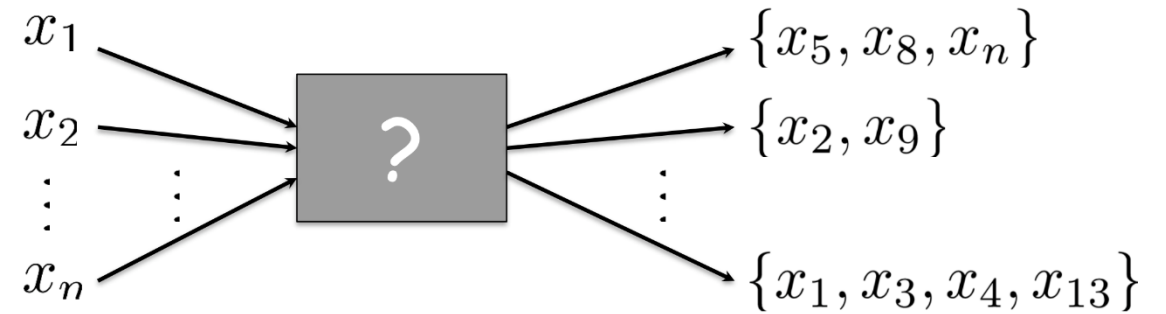
# Machine Learning



Supervised Learning



Unsupervised Learning



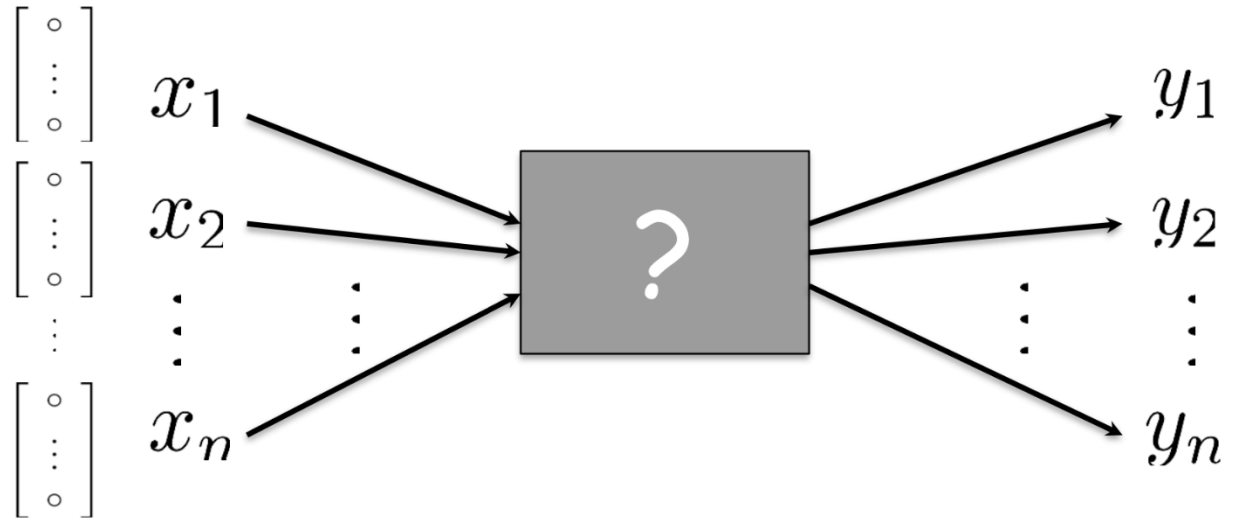
\* independent variable; predictor; feature

# dependent variable; target value

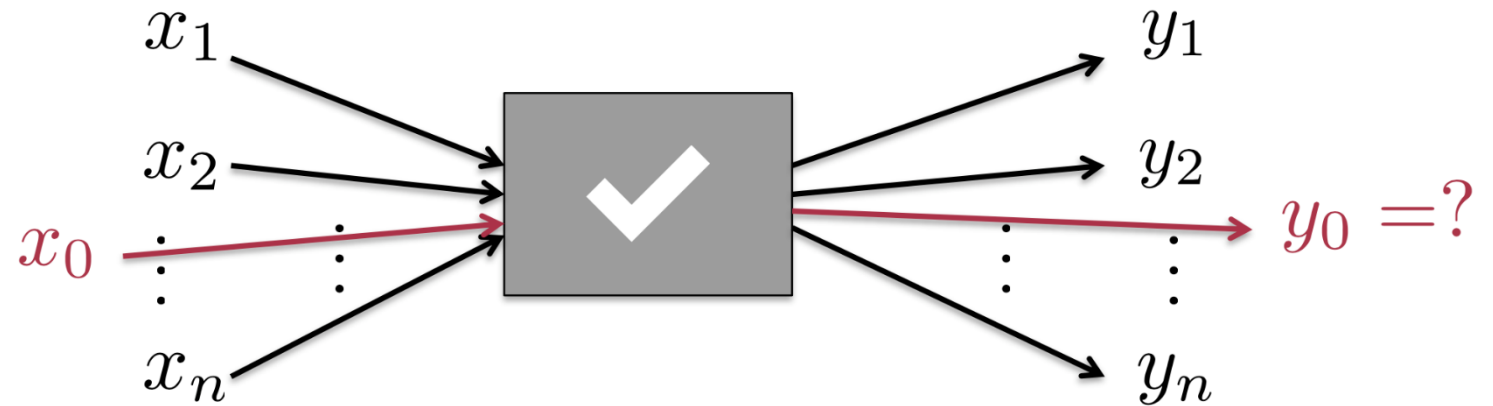


# Supervised Learning

Training data  
 $\{(x_i, y_i): i = 1, \dots, n\}$



Test data  
 $(x_0, y_0)$



# Supervised Learning

Customers

Customer 1

 $\begin{bmatrix} \circ \\ \vdots \\ \circ \end{bmatrix}$  $x_1$ 

Customer 2

 $\begin{bmatrix} \circ \\ \vdots \\ \circ \end{bmatrix}$  $x_2$ 

$\vdots$

 $\vdots$  $\vdots$  $\vdots$ 

Customer  $n$

 $\begin{bmatrix} \circ \\ \vdots \\ \circ \end{bmatrix}$  $x_n$ 

Premiums

 $y_1$ 

\$\$

 $y_2$ 

\$

 $\vdots$  $\vdots$  $\vdots$  $y_n$ 

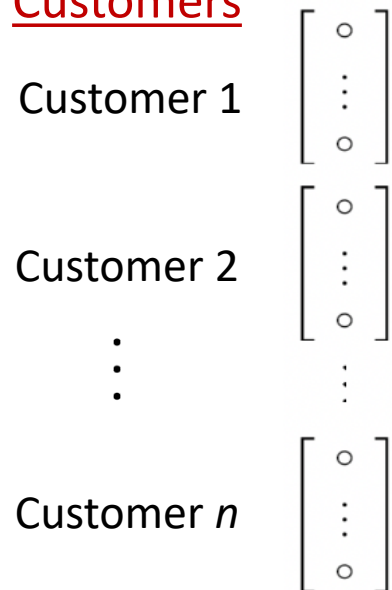
\$\$\$

Regression

Training data  
 $\{(x_i, y_i): i = 1, \dots, n\}$

# Supervised Learning

Customers

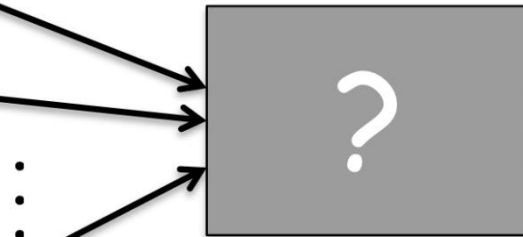


$x_1$

$x_2$

$\vdots$

$x_n$



$y_1$

$y_2$

$\vdots$

$y_n$

Claims

Valid

Fraud

$\vdots$

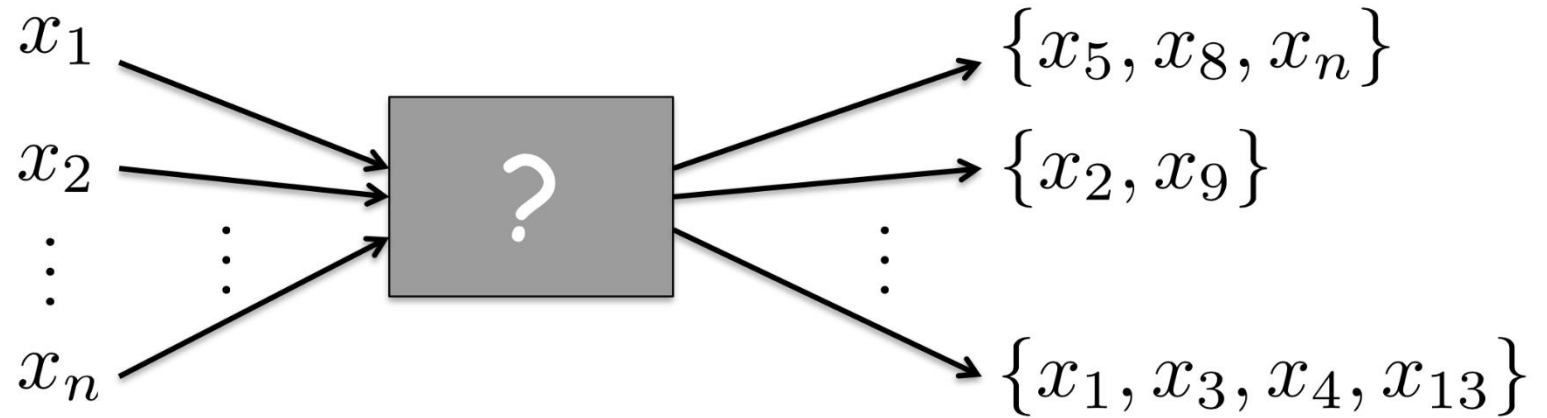
Valid

Classification

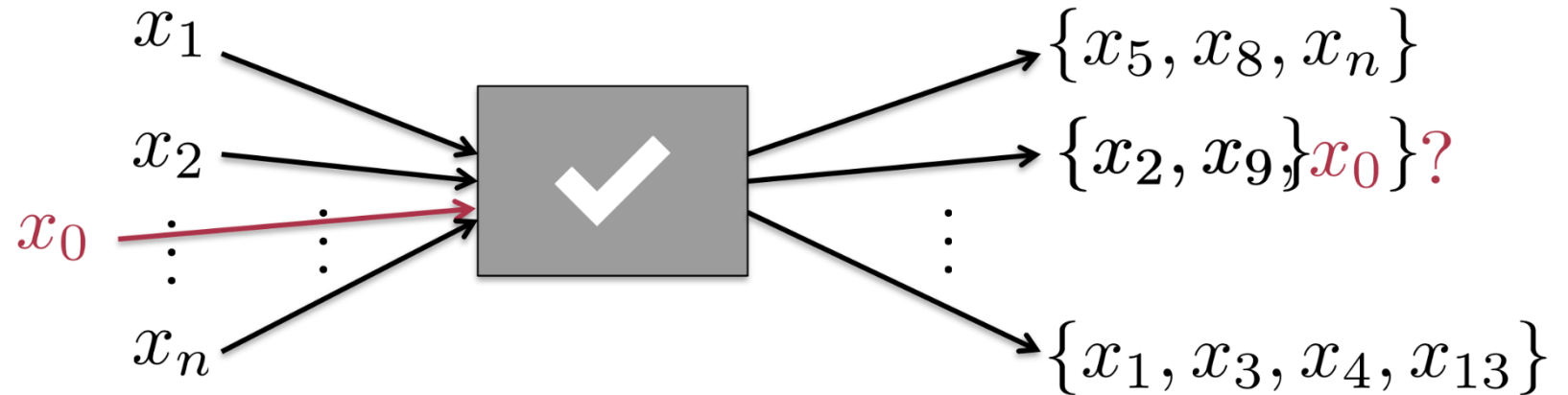
Training data  
 $\{(x_i, y_i): i = 1, \dots, n\}$

# Unsupervised Learning

Training data  
 $\{x_i : i = 1, \dots, n\}$



Test data  
 $x_0$



# Unsupervised Learning

Customers

Customer 1

$x_1$

Customer 2

$x_2$

$\vdots$

$\vdots$

$\vdots$

Customer  $n$

$x_n$



Segmentation

$\{x_5, x_8, x_n\}$

$\{x_2, x_9\}$

$\vdots$

$\vdots$

$\{x_1, x_3, x_4, x_{13}\}$

Clustering

Training data

$\{x_i: i = 1, \dots, n\}$



# Expectation from Learning

Prediction

$$x_0 = \begin{bmatrix} \circ \\ \vdots \\ \circ \end{bmatrix} \longrightarrow y_0 = ?$$

What?

Inference

$$x_0 = \begin{bmatrix} \bullet \\ \vdots \\ \bullet \end{bmatrix} \xrightarrow{?} y_0$$

How?

# The Problem of Learning



Unknown Function

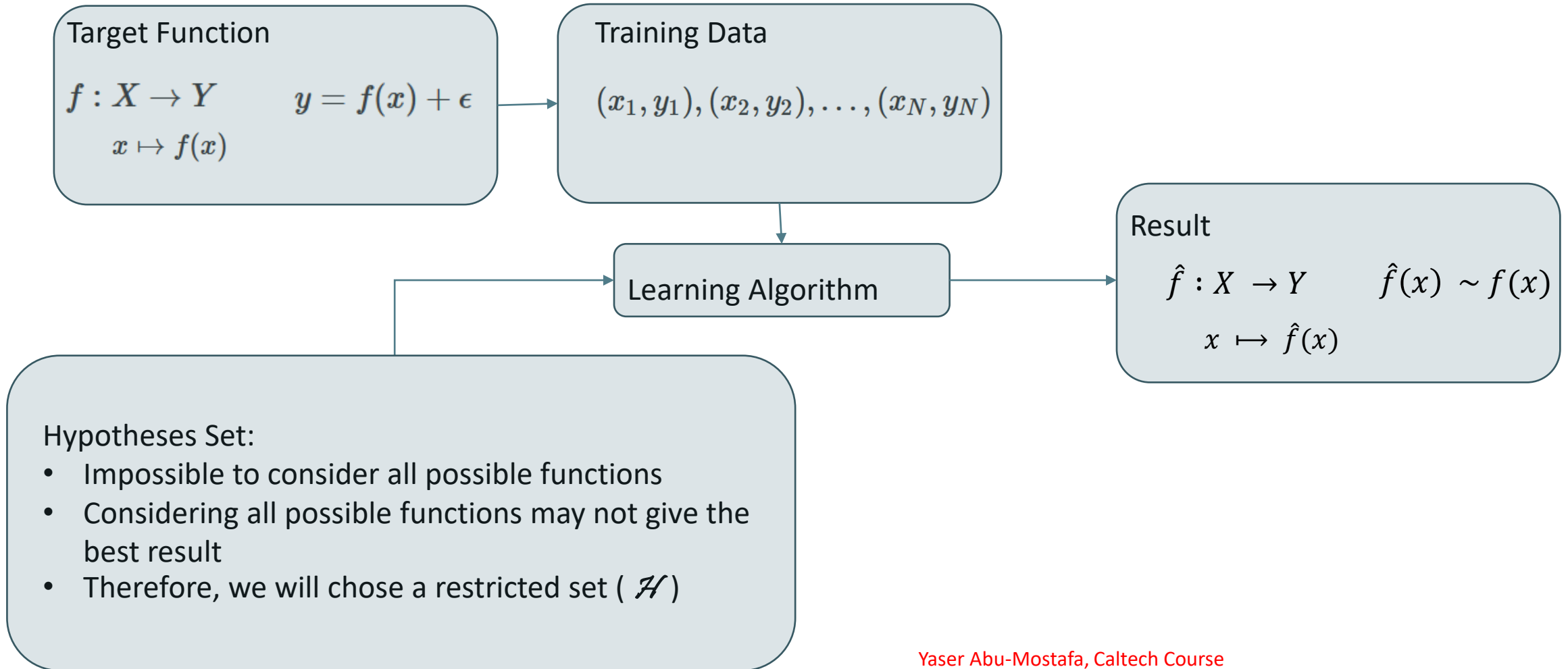
$$Y = f(X) + \epsilon$$

Random Error Term  
(Independent of Input)

Approximate?

$$\hat{Y} = \hat{f}(X)$$

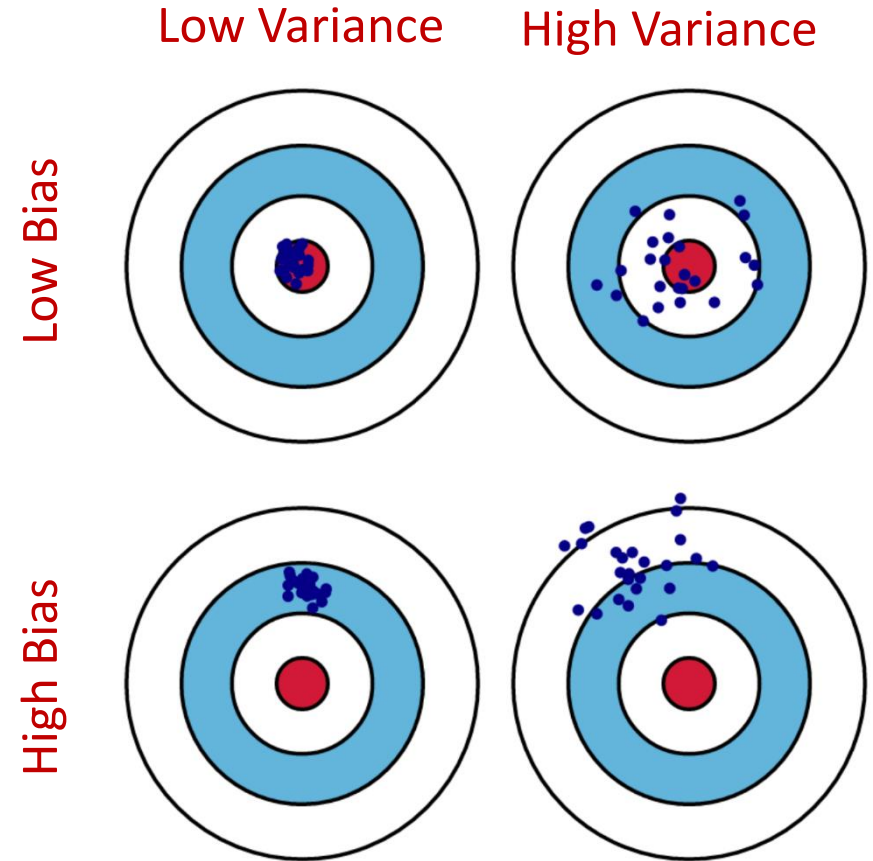
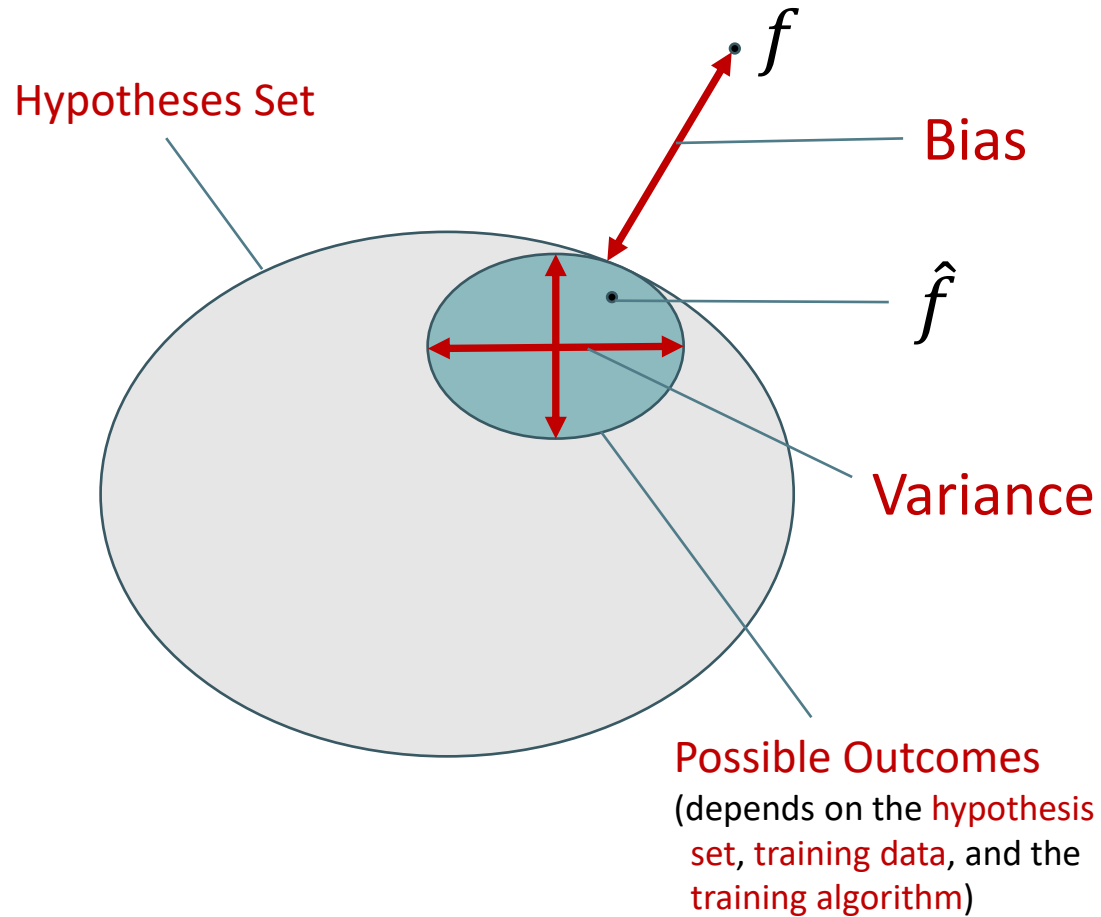
# The Problem of Learning



Yaser Abu-Mostafa, Caltech Course

<https://www.youtube.com/watch?v=mbyG85GZ0PI&list=PLD63A284B7615313A>

# The Problem of Learning



# How flexible should the model be?

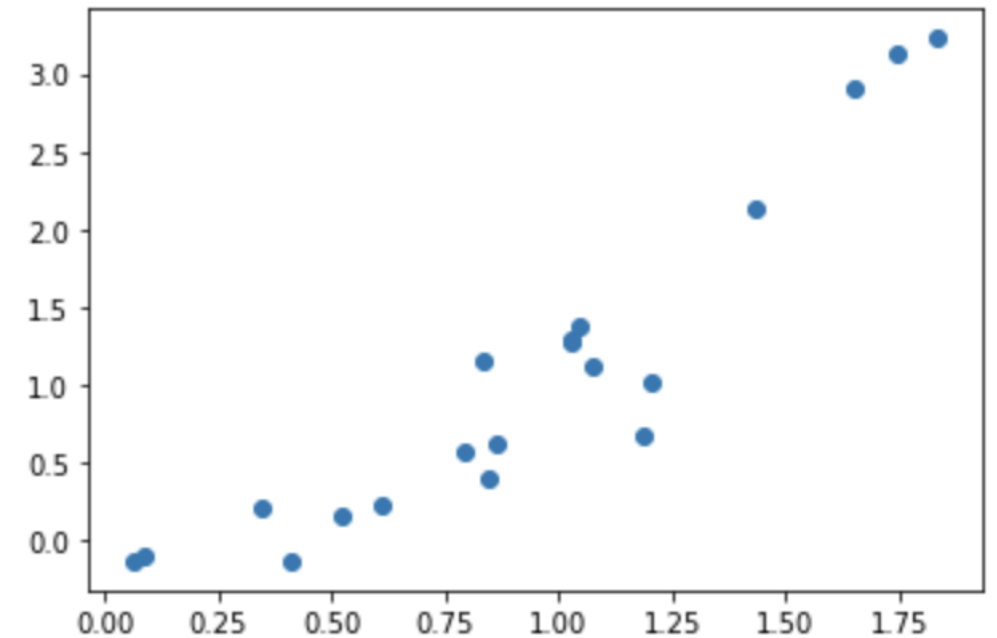
Aim: To find a function which

- Models the data best
- Performs well in the new data

Let's try a polynomial:

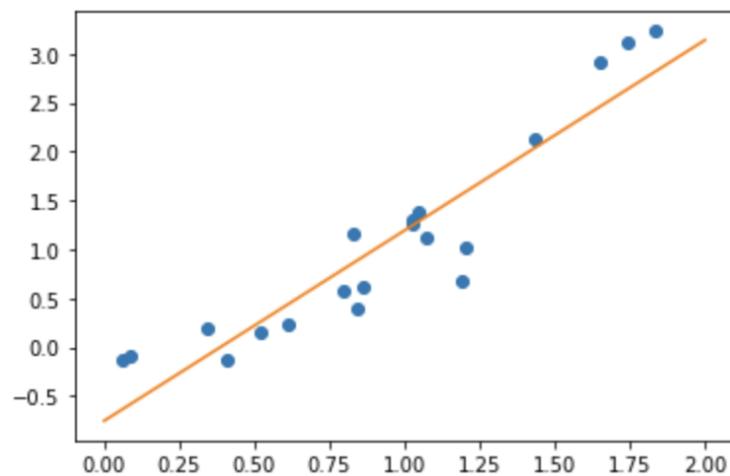
$$P(x) = c_n x^n + \dots + c_1 x + c_0$$

What should  $n$  be?    Big-Small?

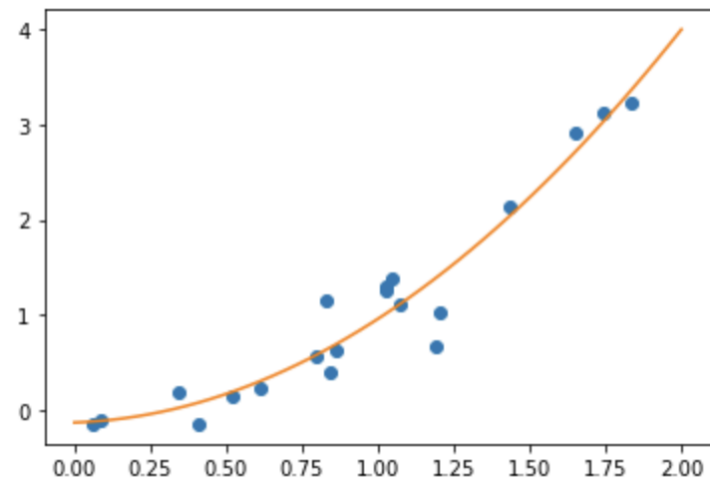




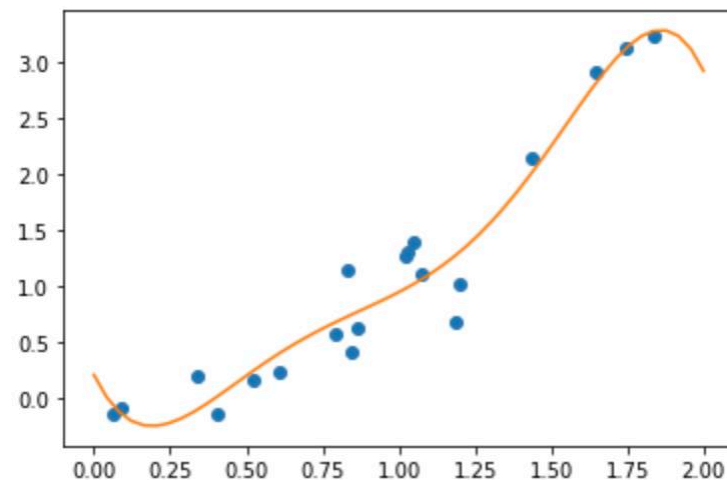
$n = 1$



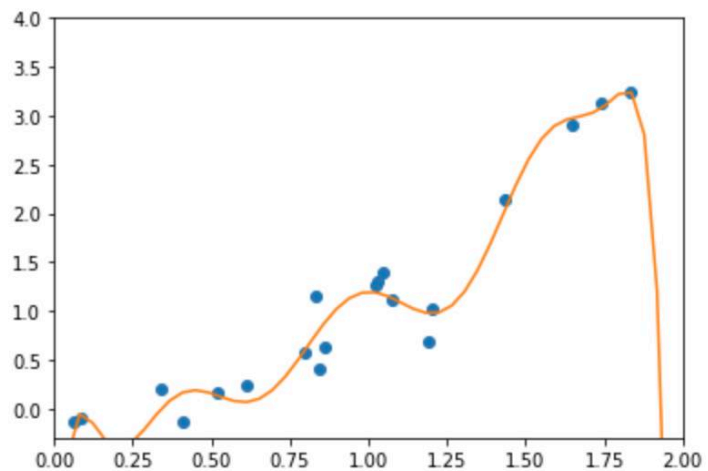
$n = 2$



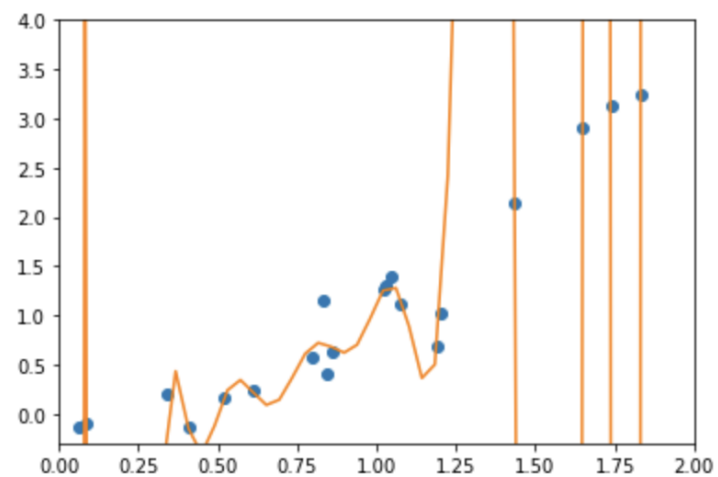
$n = 5$



$n = 10$

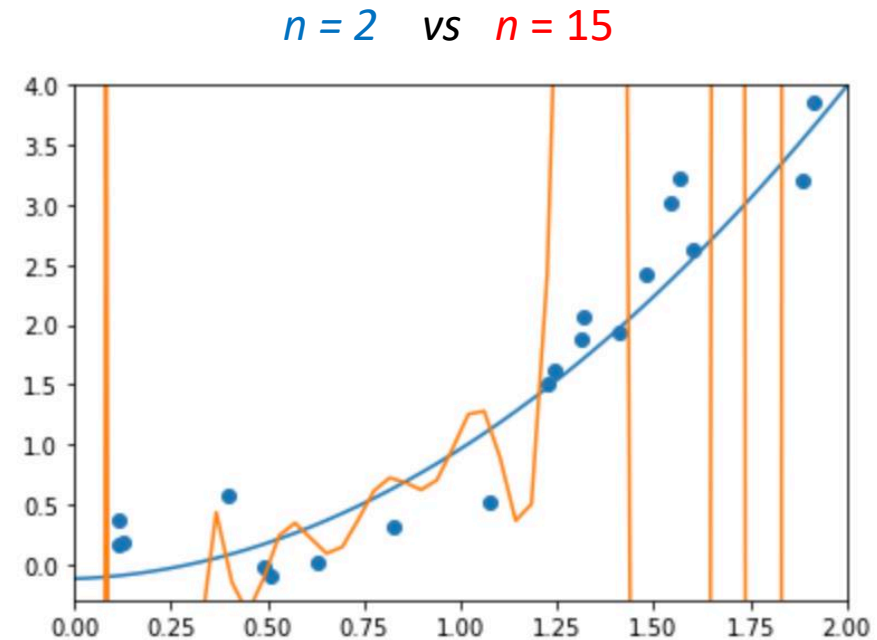
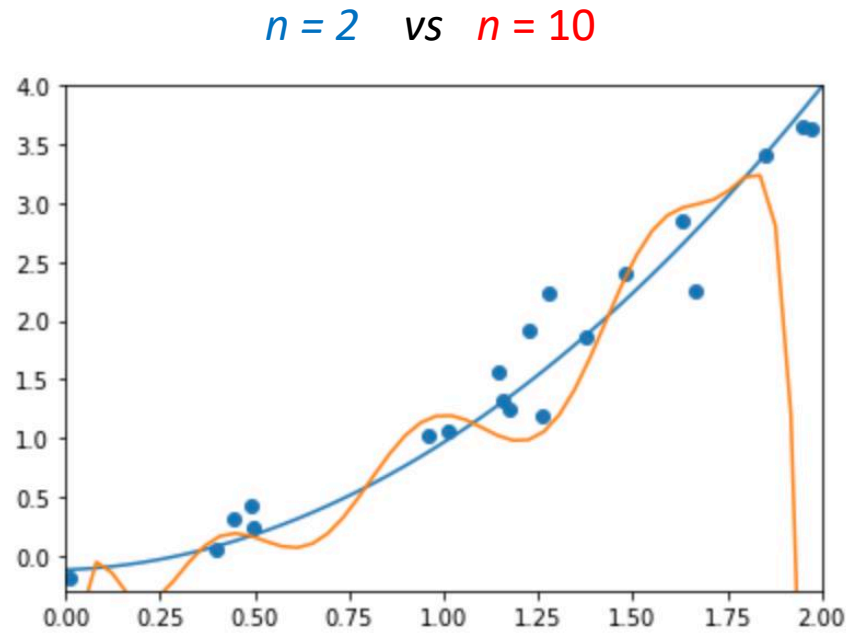


$n = 15$



This data came from a function of the form:  $Y = X^2 + \epsilon$

With new data, compare the performance of  $n = 2$  vs  $n = 10$  or  $n = 15$



Overfitting for  $n = 10, 15$ !

$n \uparrow$

Variance  $\uparrow$

Bias  $\downarrow$