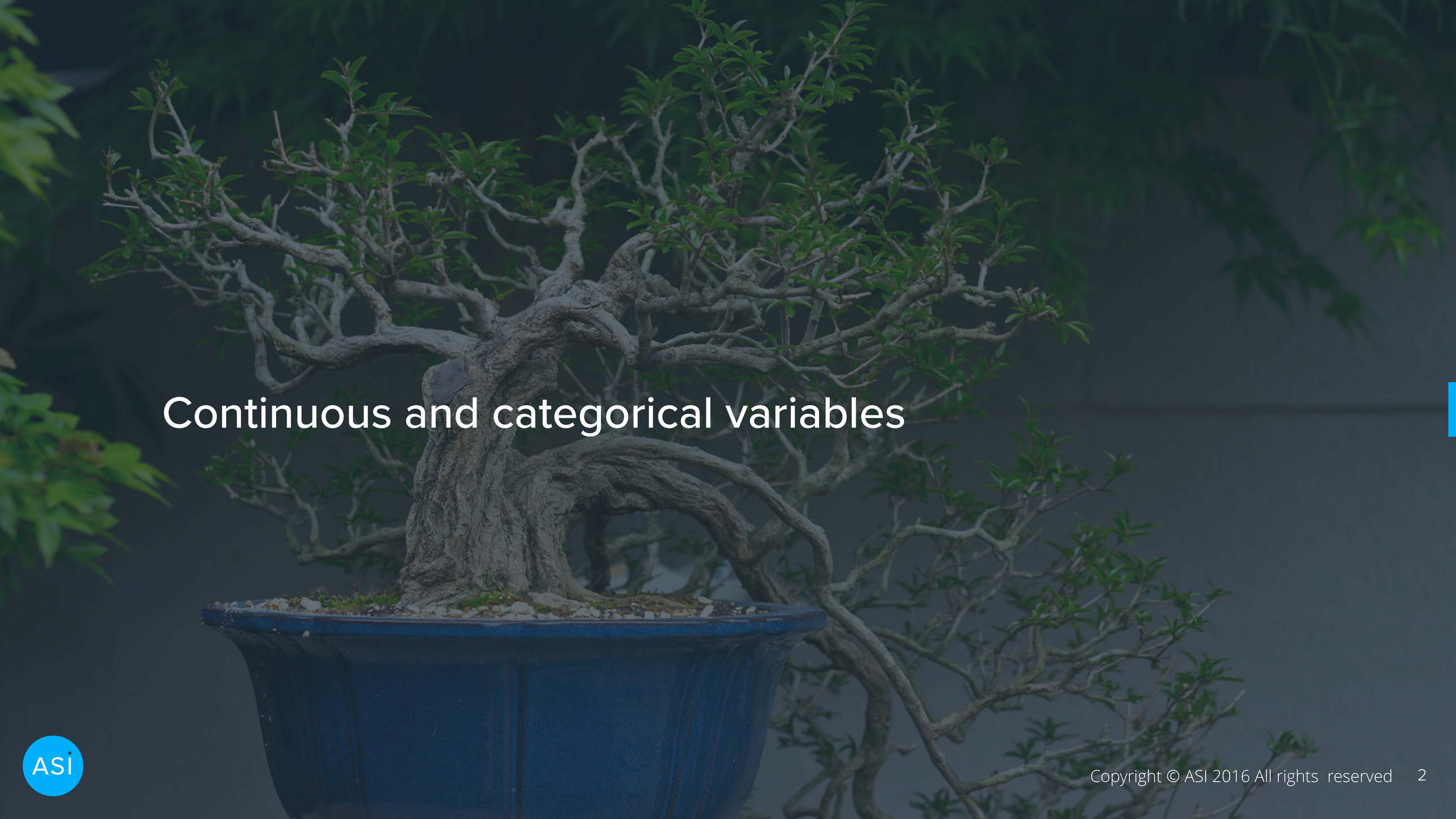
ASi

DATA SCIENCE

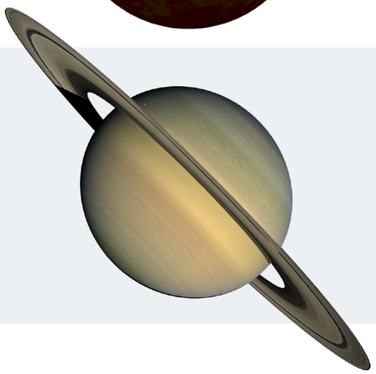
Regression and your first machine learning algorithm

Machine-learning crash course, 4th October 2017

Continuous and categorical variables



Continuous variables

PLANET	RADIUS (km)	LENGTH OF DAY (s)	NUM. MOONS
	6371	86400	1
	3390	88775	0
	58232	38517	62

Categorical variables

ANIMAL	COLOUR	SHAPE OF EARS	IS RODENT
	ORANGE	POINTY	FALSE
	GREY	ROUND	TRUE
	WHITE	FLOPPY	FALSE

REGRESSION

Predict a continuous variable

- QUEUEING TIME AT THE POST OFFICE
- GDP OF THE UK
- NUMBER OF CARS SOLD NEXT YEAR

CLASSIFICATION

Predict a categorical variable

- OUTCOME OF AN ELECTION
- APPROVAL OF A BANK LOAN
- IMAGE RECOGNITION OF STREET SIGNS

Queueing time prediction

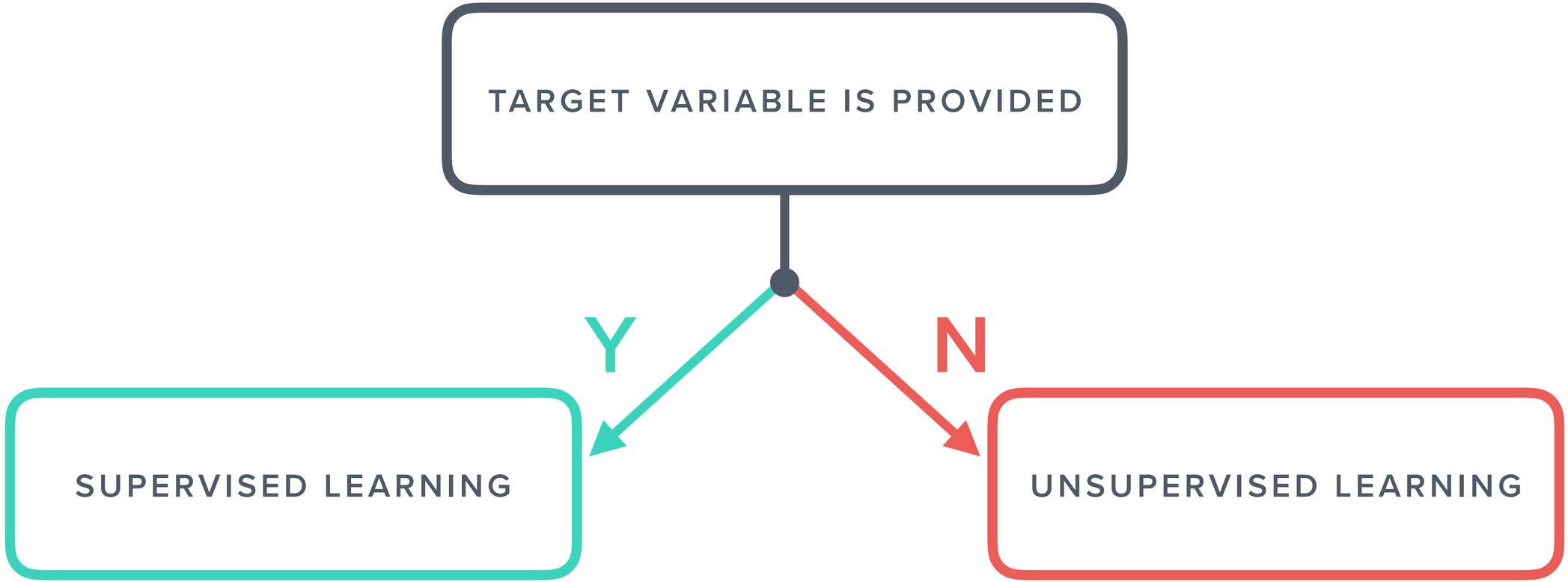
POSITION IN QUEUE	NUM. TILLS SERVING	WAITING TIME (min)
8	4	43
6	3	31
10	4	68

Features

X

Target

y



Session 2 (**Today**)

Session 3 (**Random Forests**)

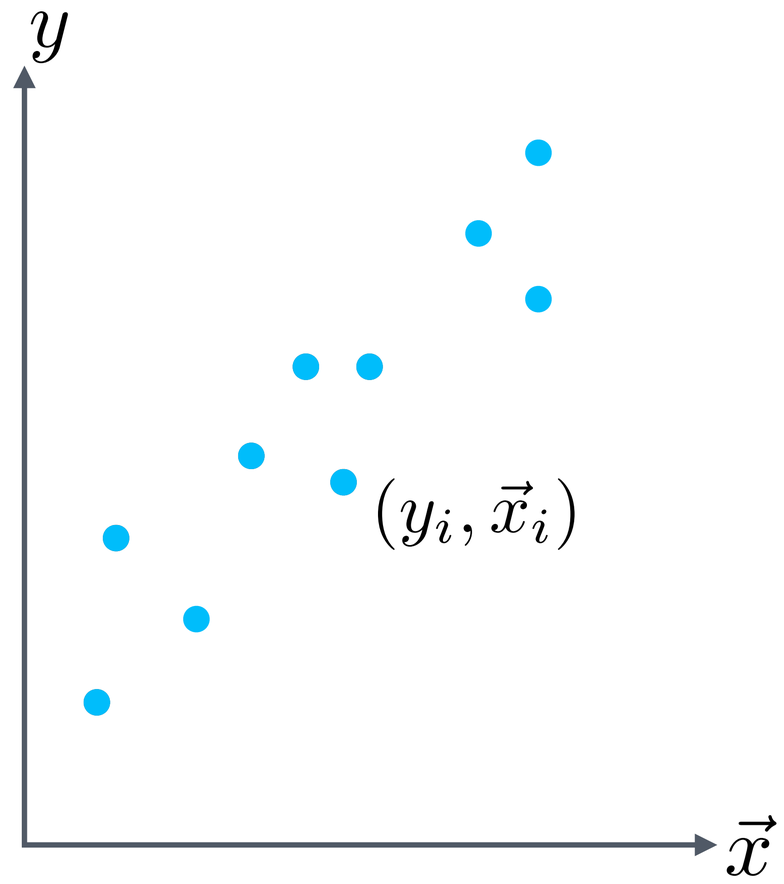
Session 5 (**Neural Networks**)

Session 4 (**Clustering**)

Regression - Curve Fitting

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER



The **feature vector** reads:

$$\vec{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Mi})$$

$$i = 1, \ldots, N$$

N is the **number of datapoints**.

M is the **number of features**.

ASI

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

Estimate of the target variable

Some function (Potentially non-linear)

$$\hat{y}_i = f\left(\vec{x}_i, \vec{\theta}\right) = f\left(\vec{x}_i, \theta_0, \theta_1, \ldots, \theta_{C-1}\right)$$

Vector of parameters

Model complexity (Number of parameters)

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

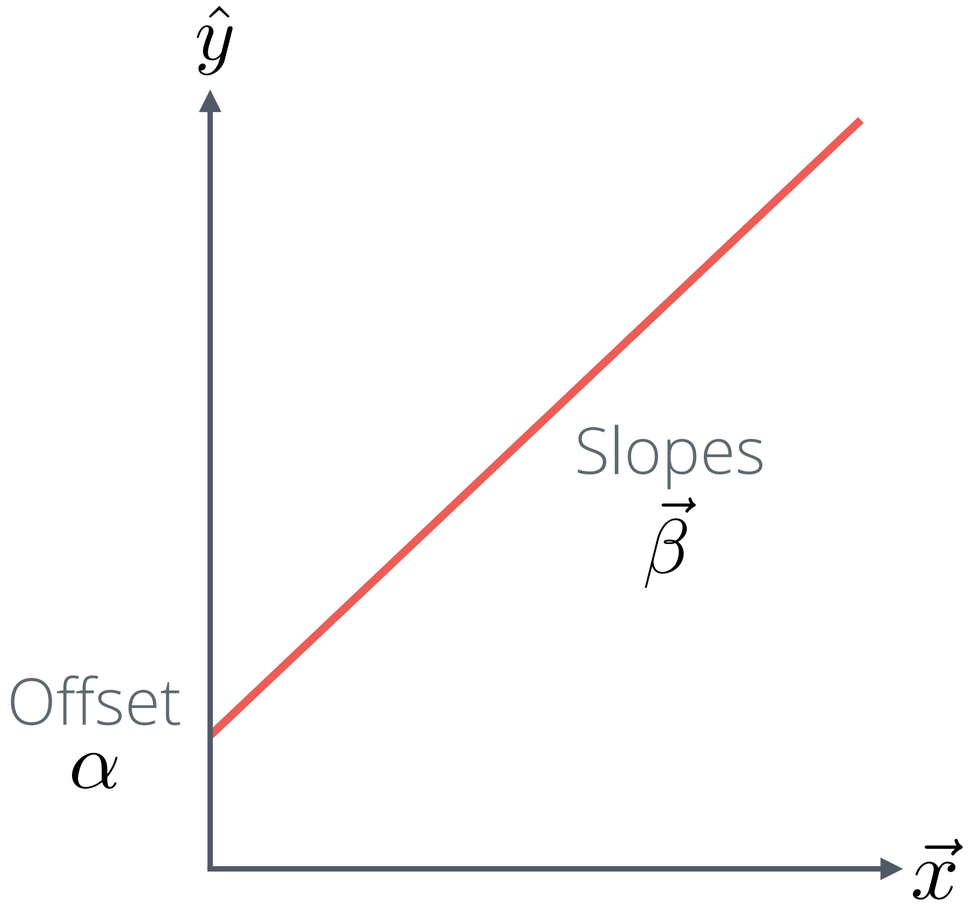
LINEAR REGRESSION

$$\hat{y}_i = f(\vec{x}_i, \vec{\theta}) = \alpha + \vec{\beta} \cdot \vec{x}_i$$

$$\alpha = \theta_0$$

$$\vec{\beta} = (\theta_1, \theta_2, \ldots, \theta_{C-1})$$

$$M = C - 1$$



ASI

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

The **cost function** is equal to minus the logarithm of the likelihood function.

The **likelihood function** is the probability of the data given the parameters.

Select the parameter values that **maximise** the likelihood function.

The same parameter values **minimise** the cost function.

Making use of the cost function improves **numerical stability**.

The procedure of finding optimal parameters is also called **training**.

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

Cost function

Likelihood function

$$J(\vec{\theta}) = -\log p\left(\text{Datapoints}\middle|\vec{\theta}\right)$$

$$\hat{\vec{\theta}} = \underset{\vec{\theta}}{\operatorname{argmin}}\, J(\vec{\theta}) = \underset{\vec{\theta}}{\operatorname{argmax}}\, p\left(\text{Datapoints}\middle|\vec{\theta}\right)$$

'Trained' parameters

Minimum cost

Maximum likelihood

We need to specify the form of the cost function!

ASI

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

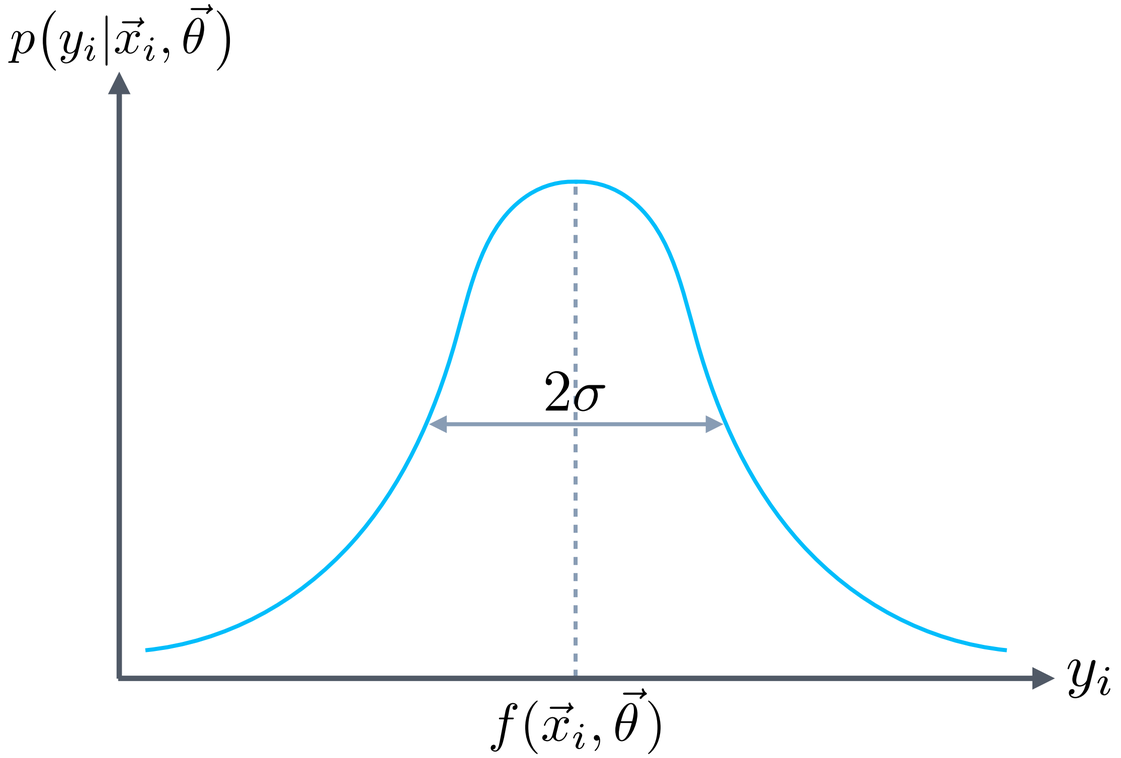
LEAST SQUARES

1. Datapoints are **independent**:

$$p(\text{Datapoints}|\vec{\theta}) = \prod_{i=1}^{N} p\left(y_i|\vec{x}_i,\vec{\theta}\right)$$

2. Target variable is **normal-distributed**:

$$p\left(y_i|\vec{x}_i,\vec{\theta}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(y_i - f(\vec{x}_i,\vec{\theta})\right)^2}{2\sigma^2}\right]$$



ASI

REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

LEAST SQUARES

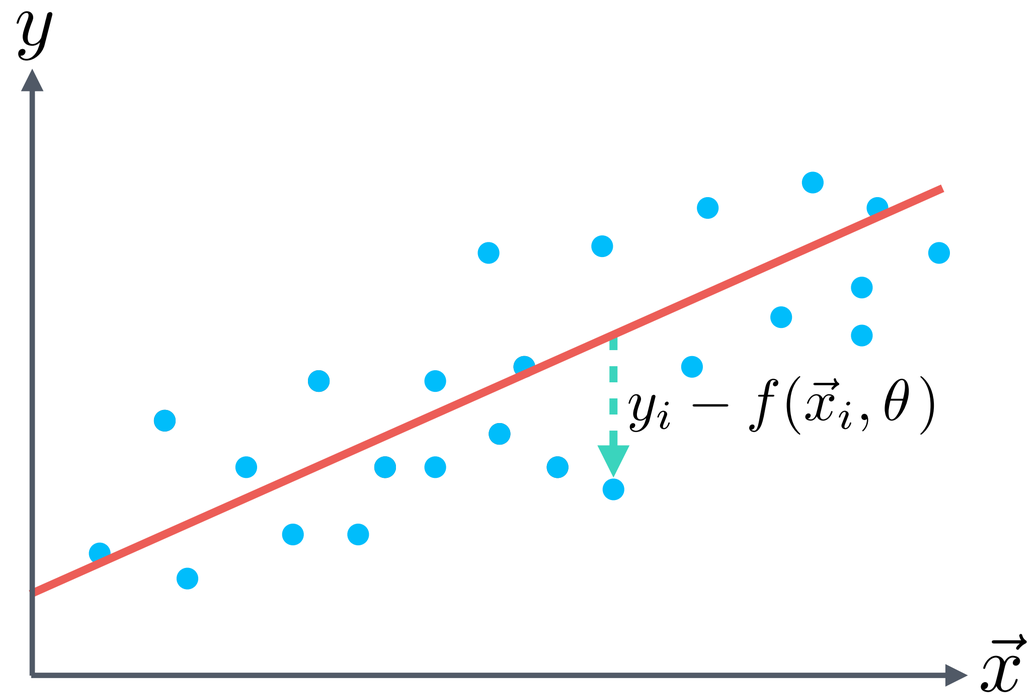
The cost function is found to be:

$$J(\vec{\theta}) = \frac{N}{2}\left(\frac{K(\vec{\theta})}{\sigma^2} + \log(2\pi\sigma^2)\right)$$

$$K(\vec{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(\vec{x}_i, \vec{\theta})\right)^2$$

Cost minimisation simplifies to:

$$\hat{\vec{\theta}} = \underset{\vec{\theta}}{\operatorname{argmin}}\, K(\vec{\theta})$$

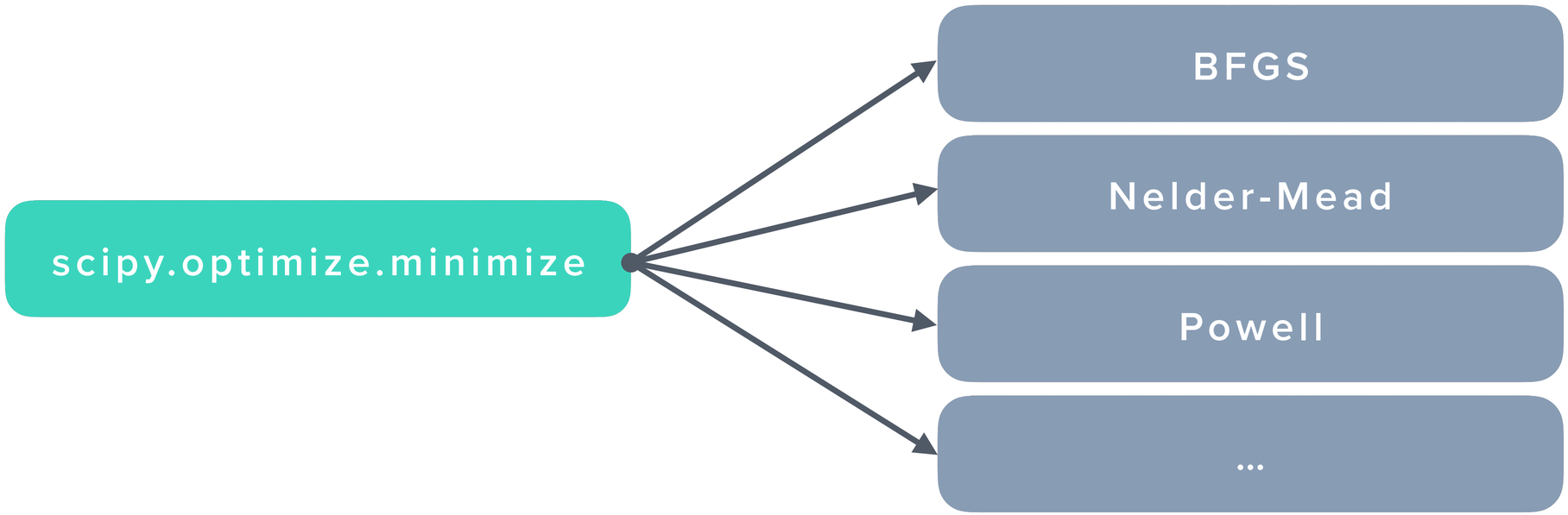


ASI

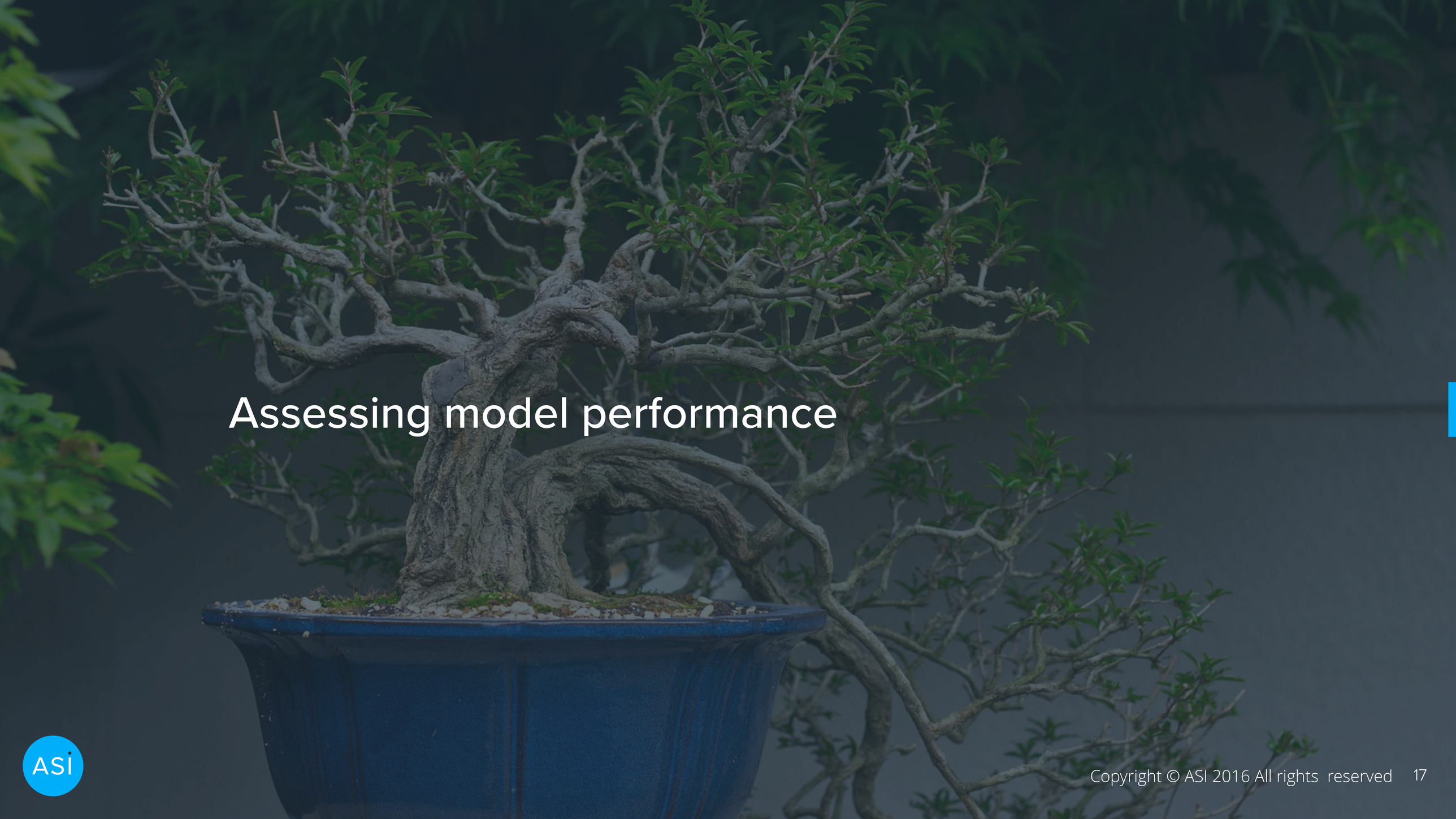
REGRESSION

DATAPOINTS | MODEL | COST FUNCTION | OPTIMISER

To find the ‘best’ parameter values we need an **optimisation algorithm**.
It is typically a good idea to use a pre-packaged implementation.



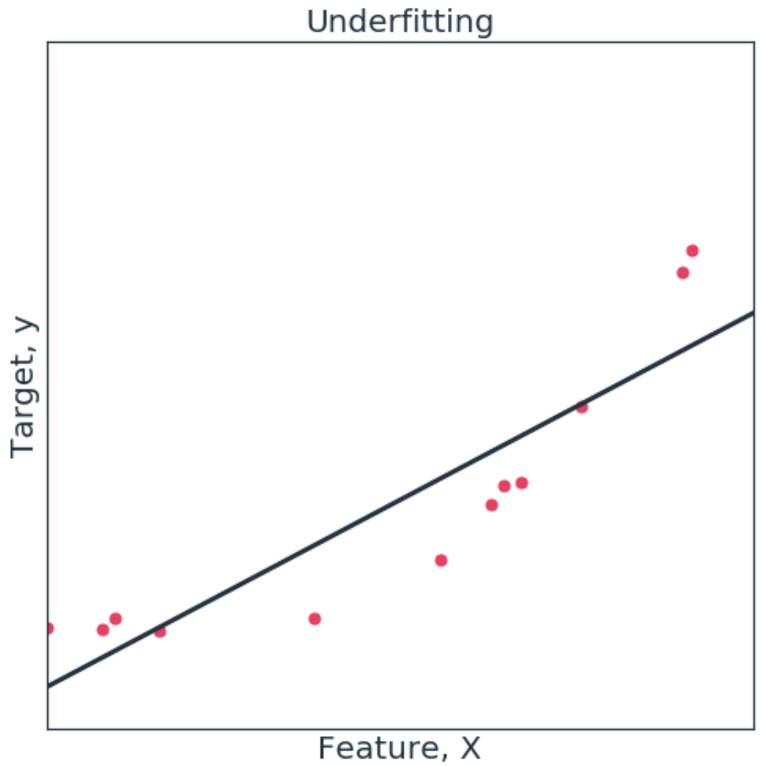
ASI



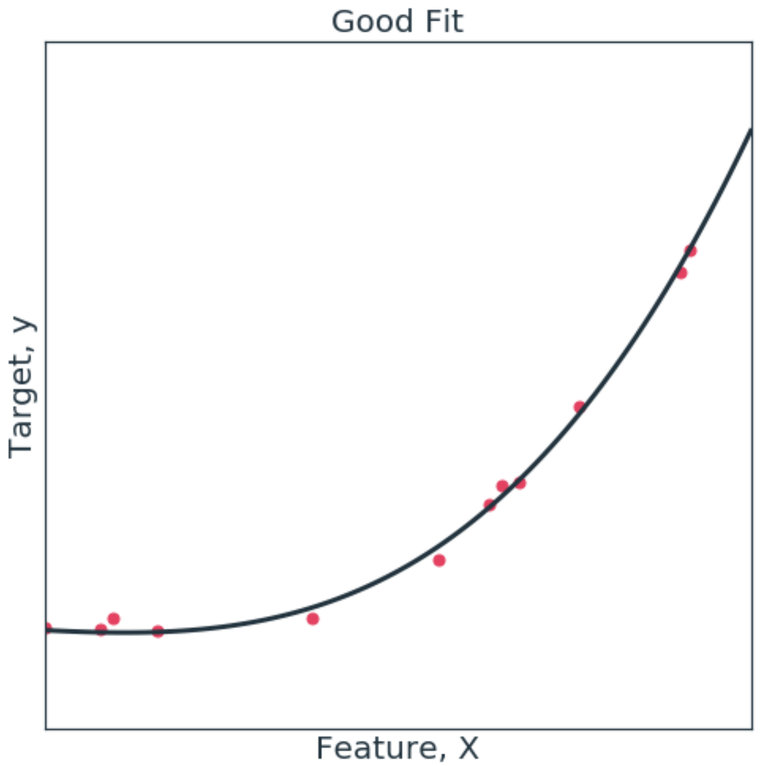
Assessing model performance

Underfitting (High Bias) and Overfitting (High Variance)

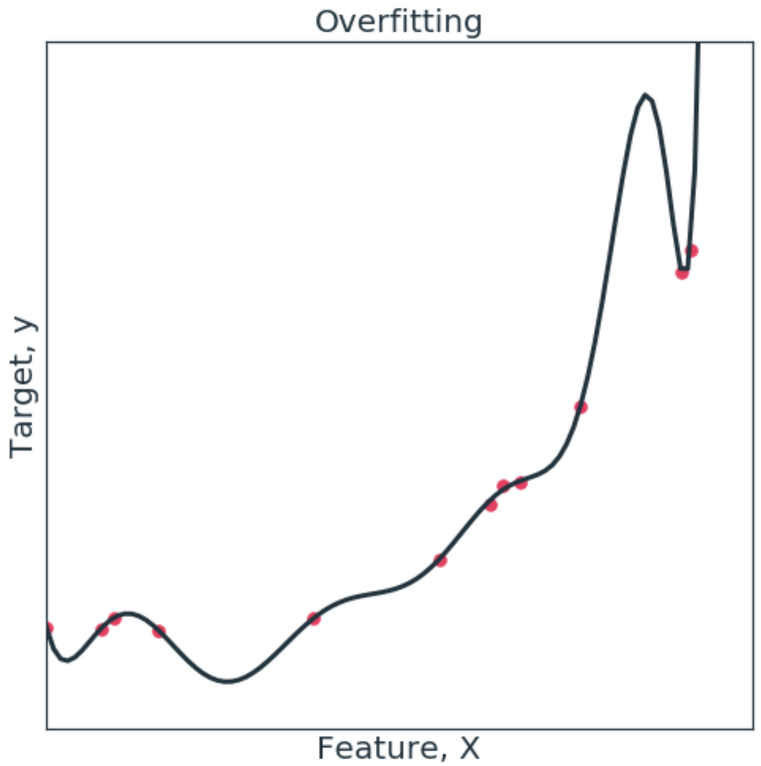
The two most common reasons why a machine-learning algorithm has poor performance.



Too few parameters



Just right

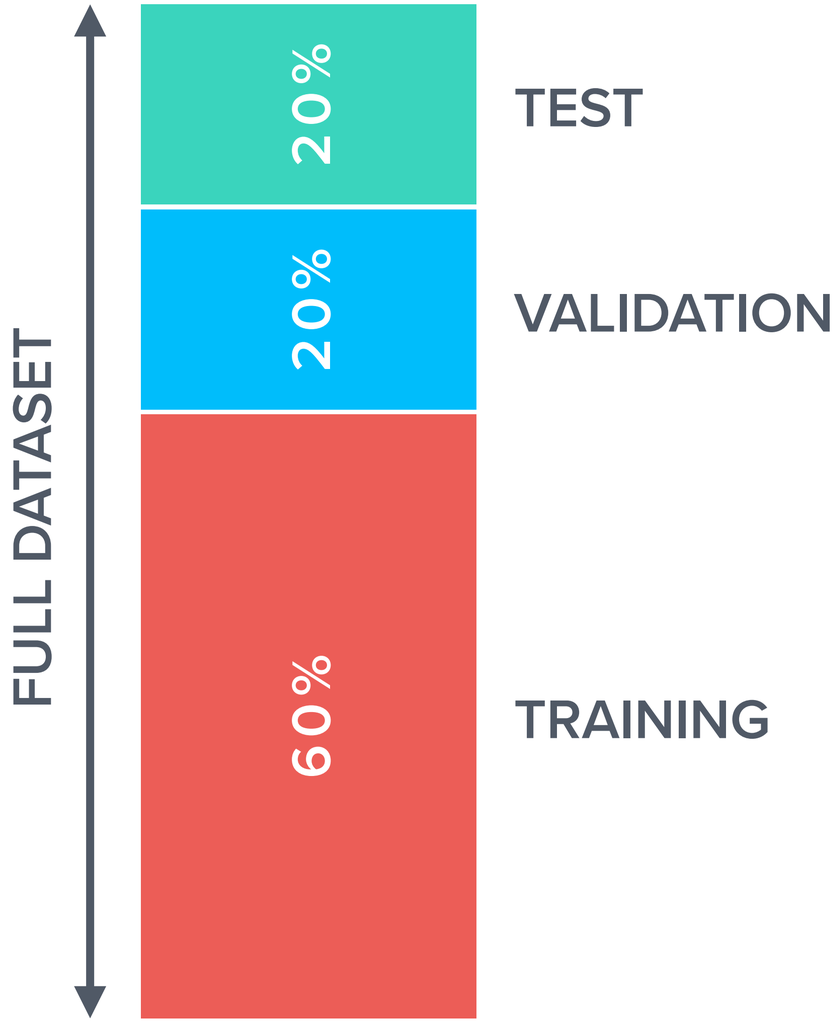


Too many parameters

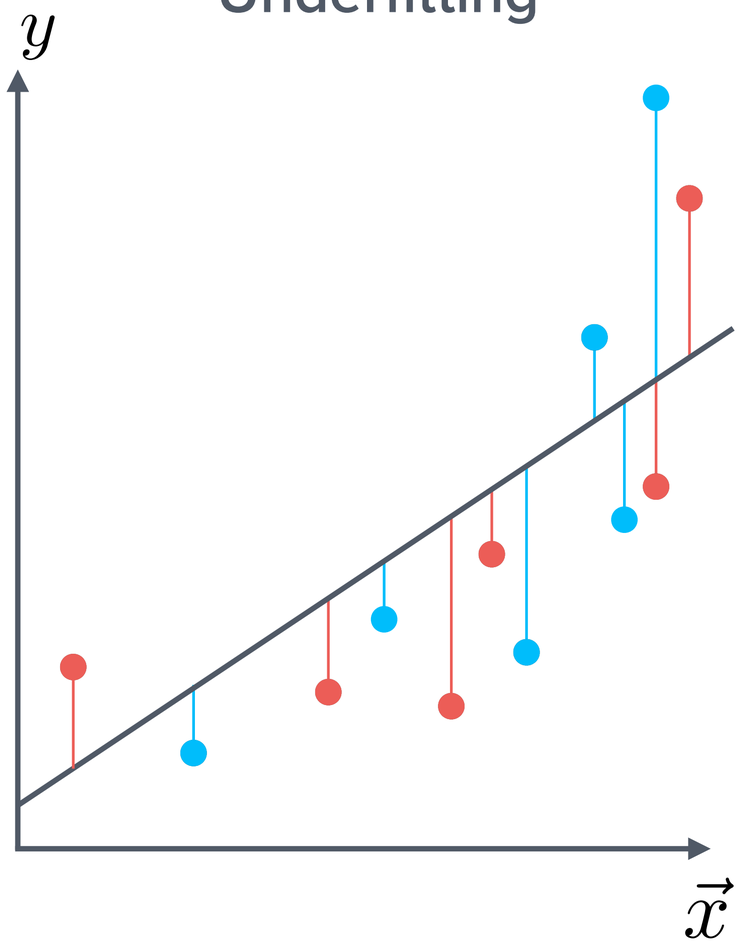
Three-way split assessment

Reliable machine-learning algorithms **generalise** well to new data.

Splitting the dataset in **training**, **validation** and **test** parts allows one to assess reliability.



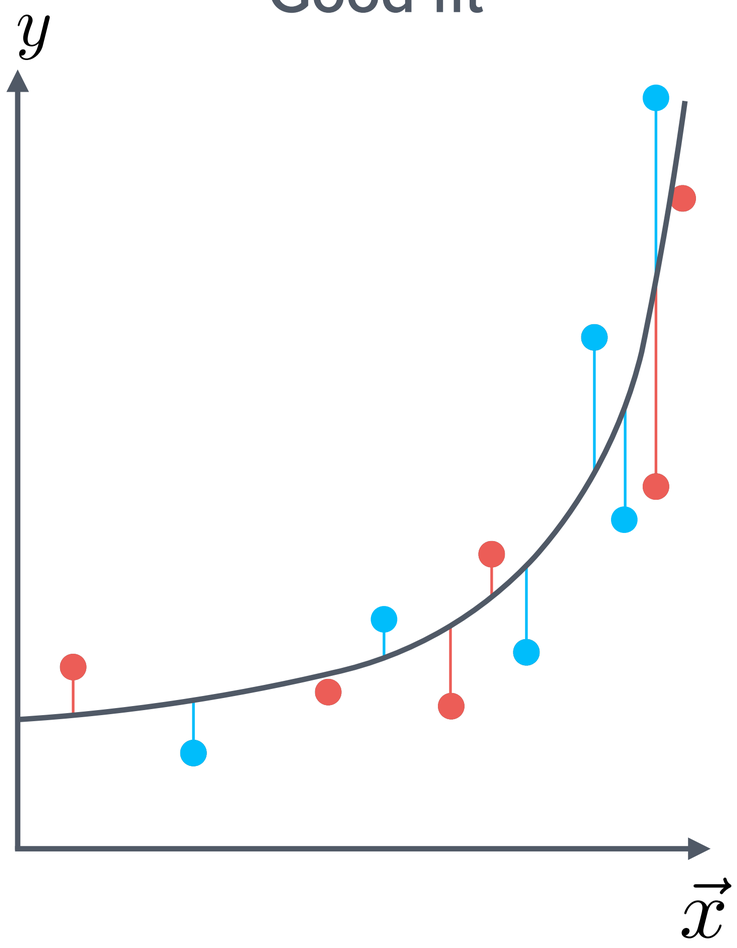
Underfitting



High training error

High validation error

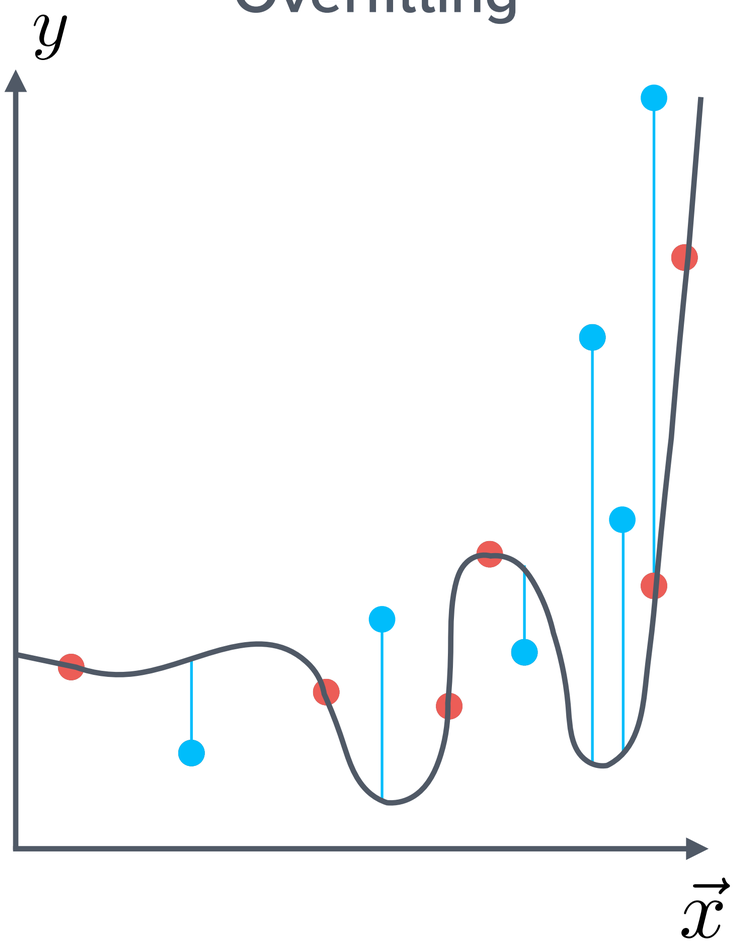
Good fit



Low training error

Low validation error

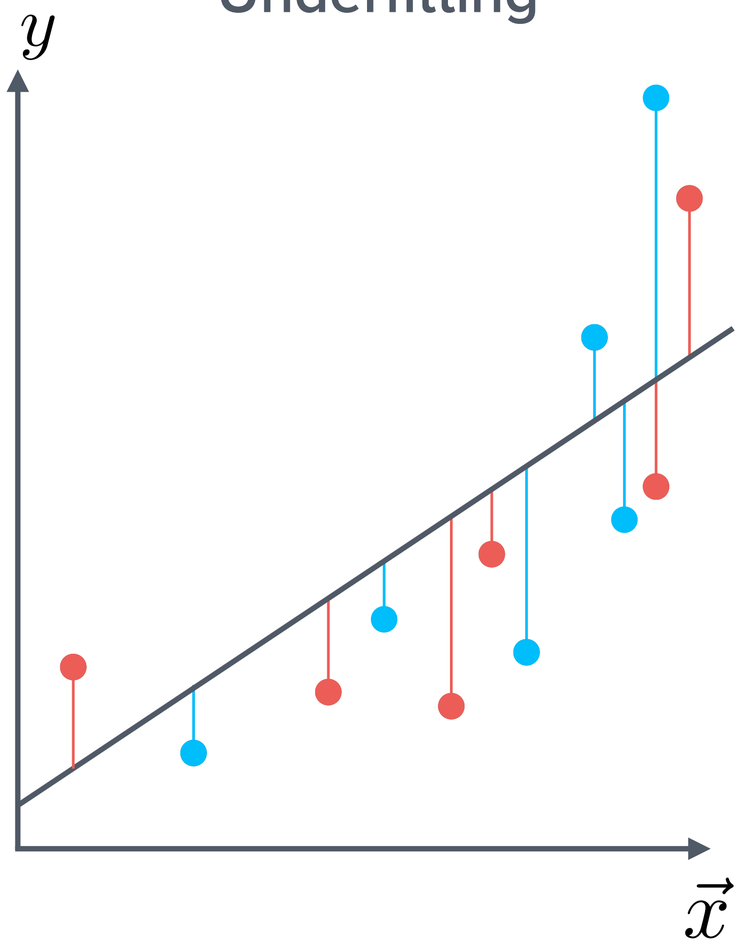
Overfitting



Low training error

High validation error

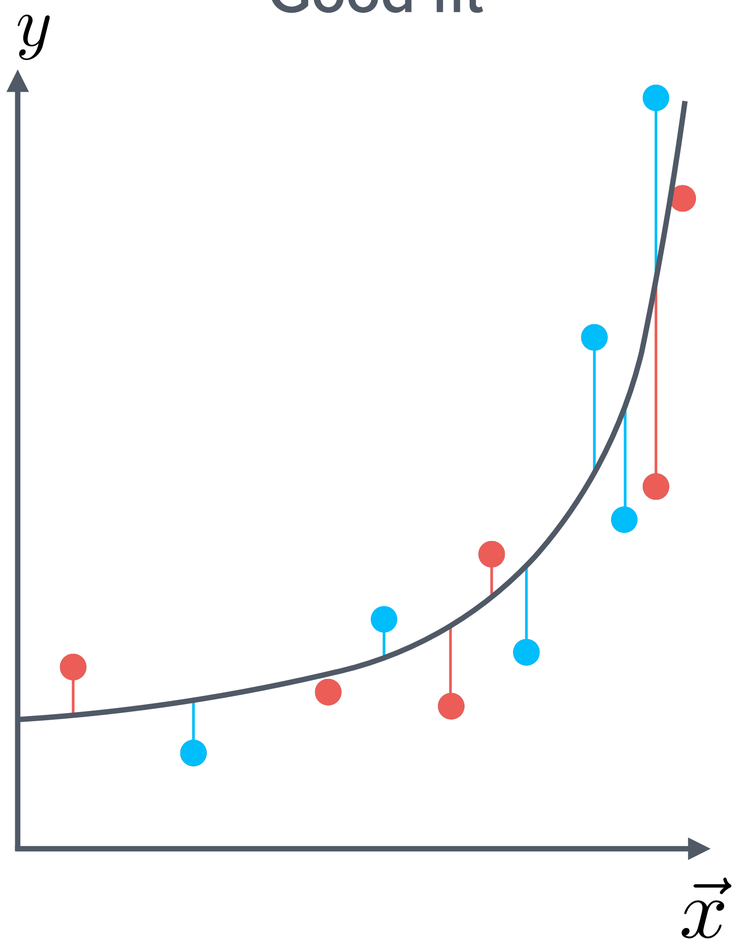
Underfitting



High training error

High validation error

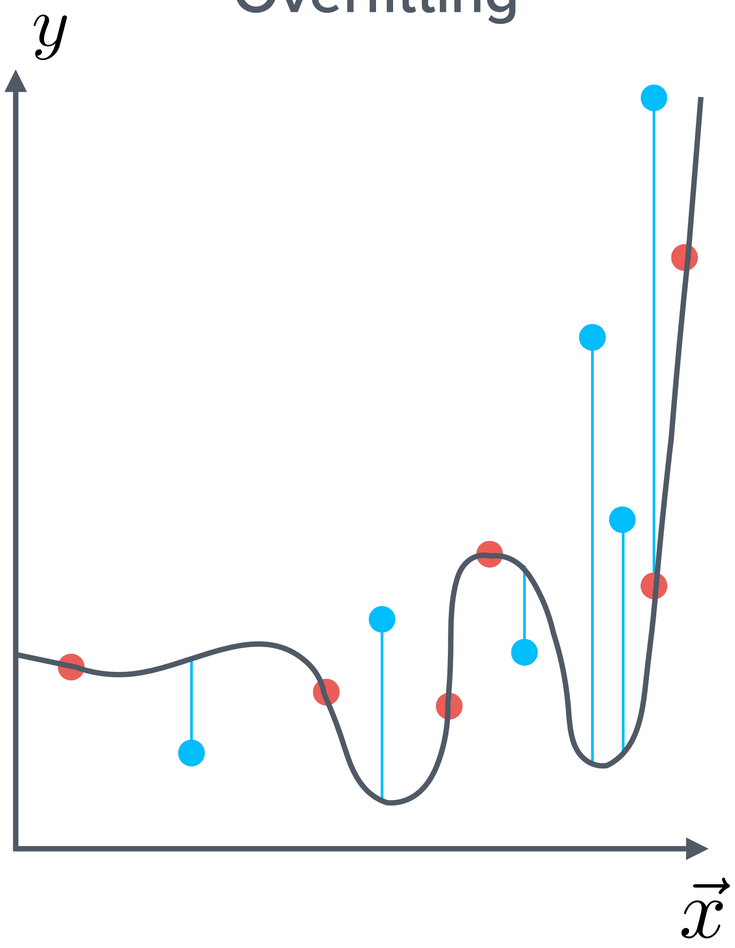
Good fit



Low training error

Low validation error

Overfitting



Low training error

High validation error

This is what matters

Three versions of the cost function

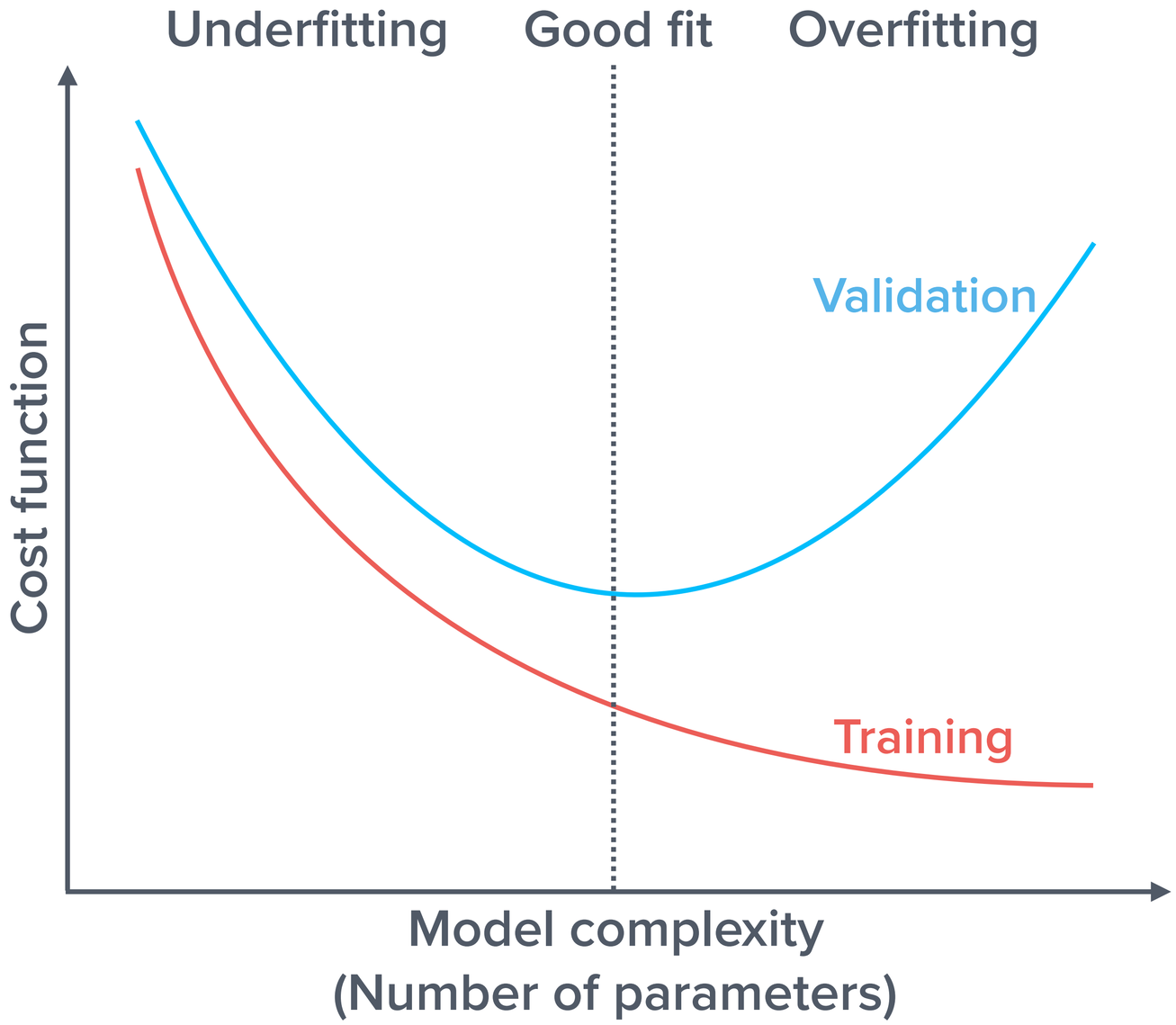
$$J(\vec{\theta}) = -\log p\left(\text{Datapoints}\middle|\vec{\theta}\right)$$

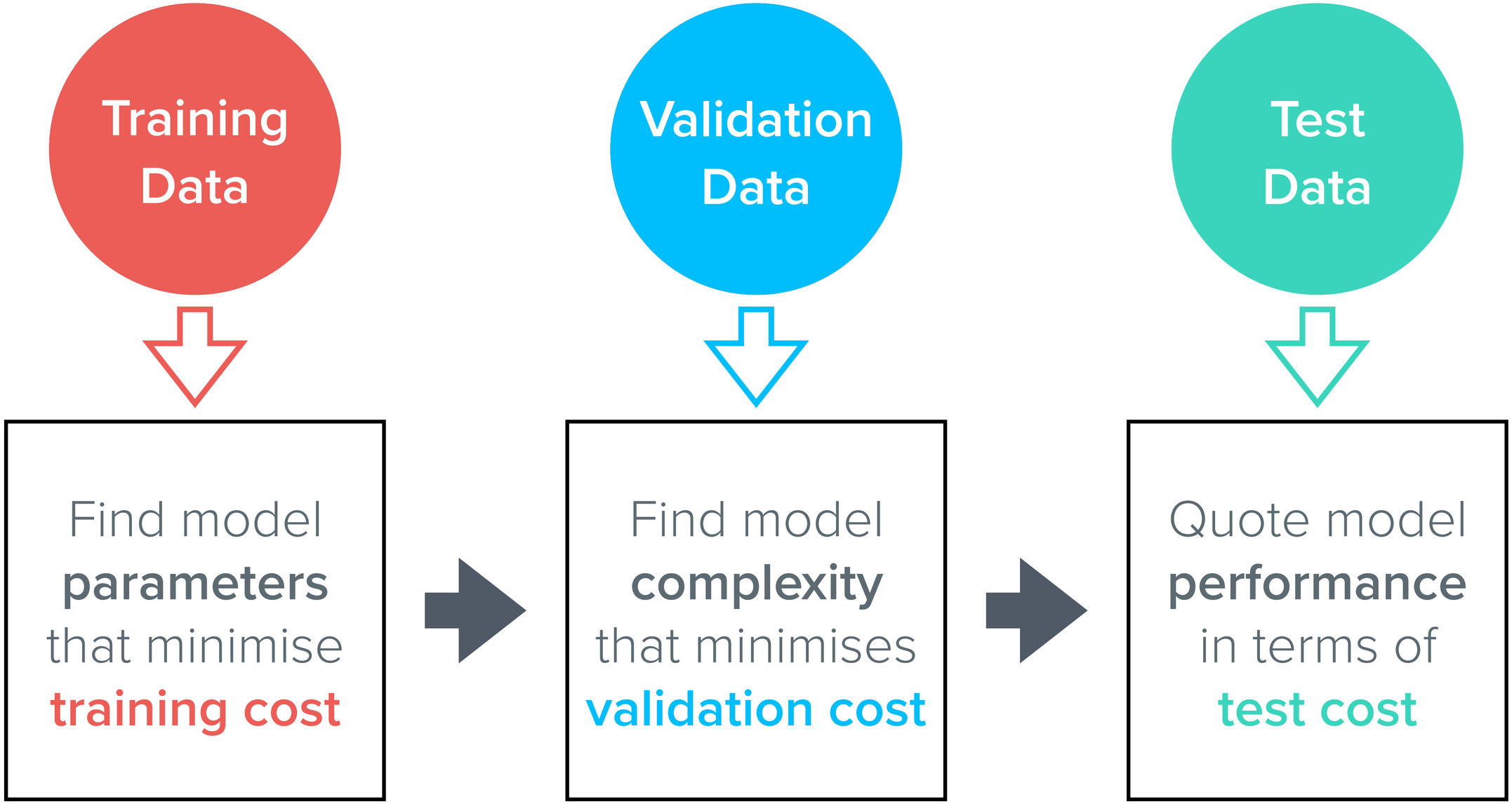
Training datapoints ⇒ **Training** cost

Validation datapoints ⇒ **Validation** cost

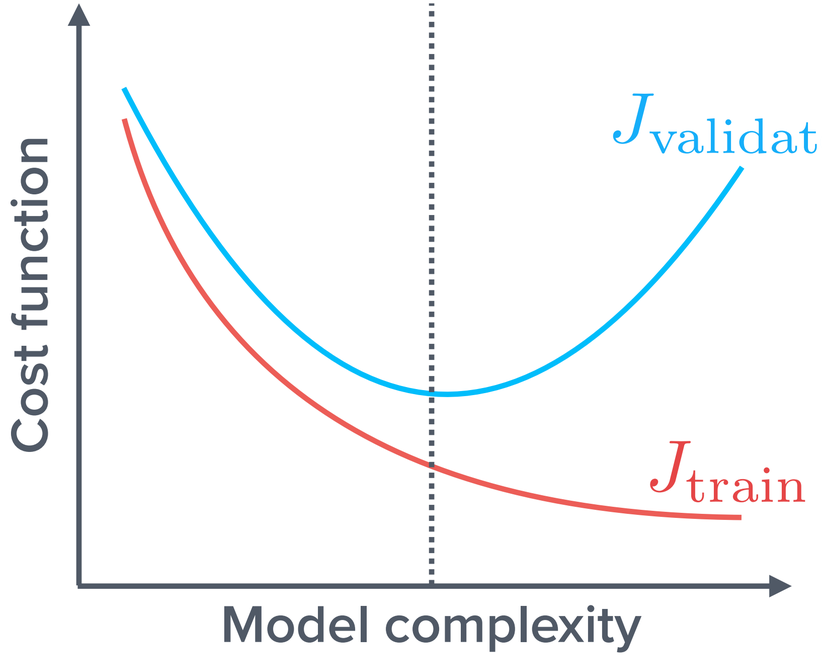
Test datapoints ⇒ **Test** cost

Bias-variance tradeoff





Diagnosing underfitting vs overfitting #1



Underfitting

J_{train} is high

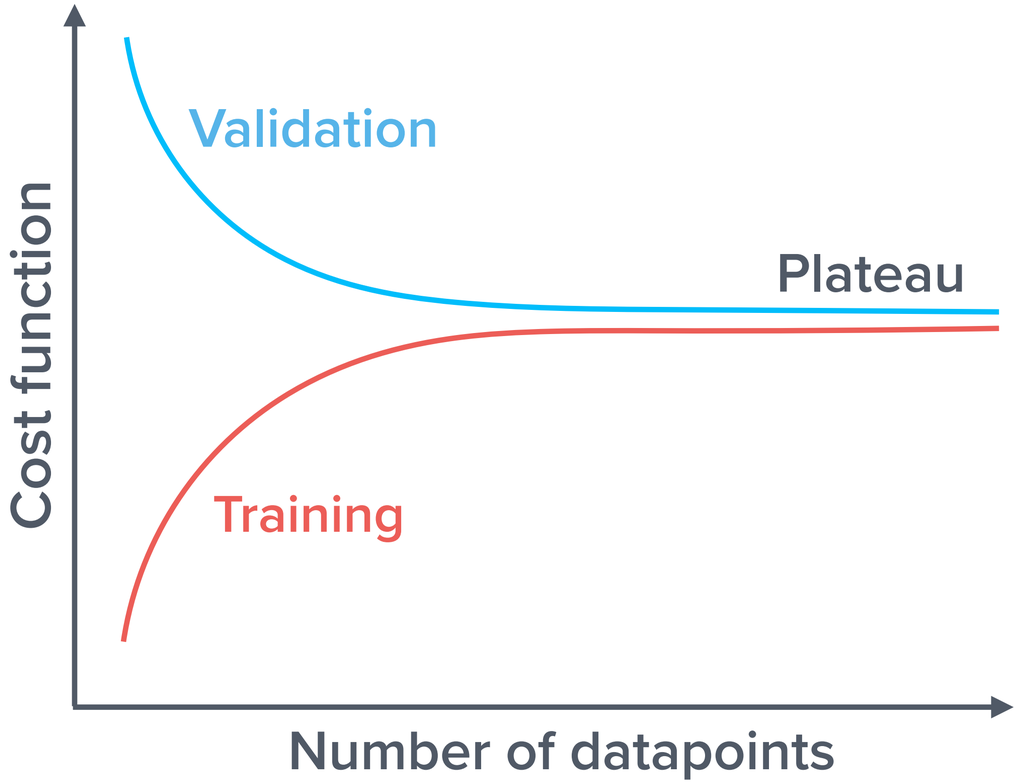
$J_{\text{train}} \approx J_{\text{validate}}$

Overfitting

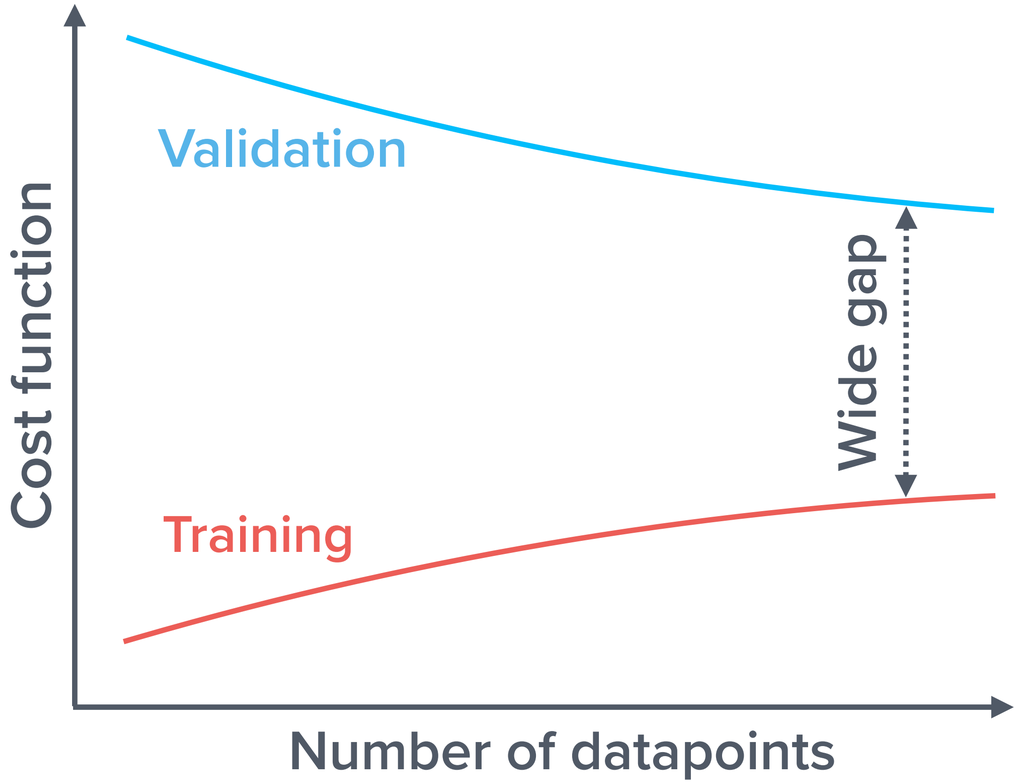
J_{train} is low

$J_{\text{train}} \ll J_{\text{validate}}$

Diagnosing underfitting vs overfitting #2

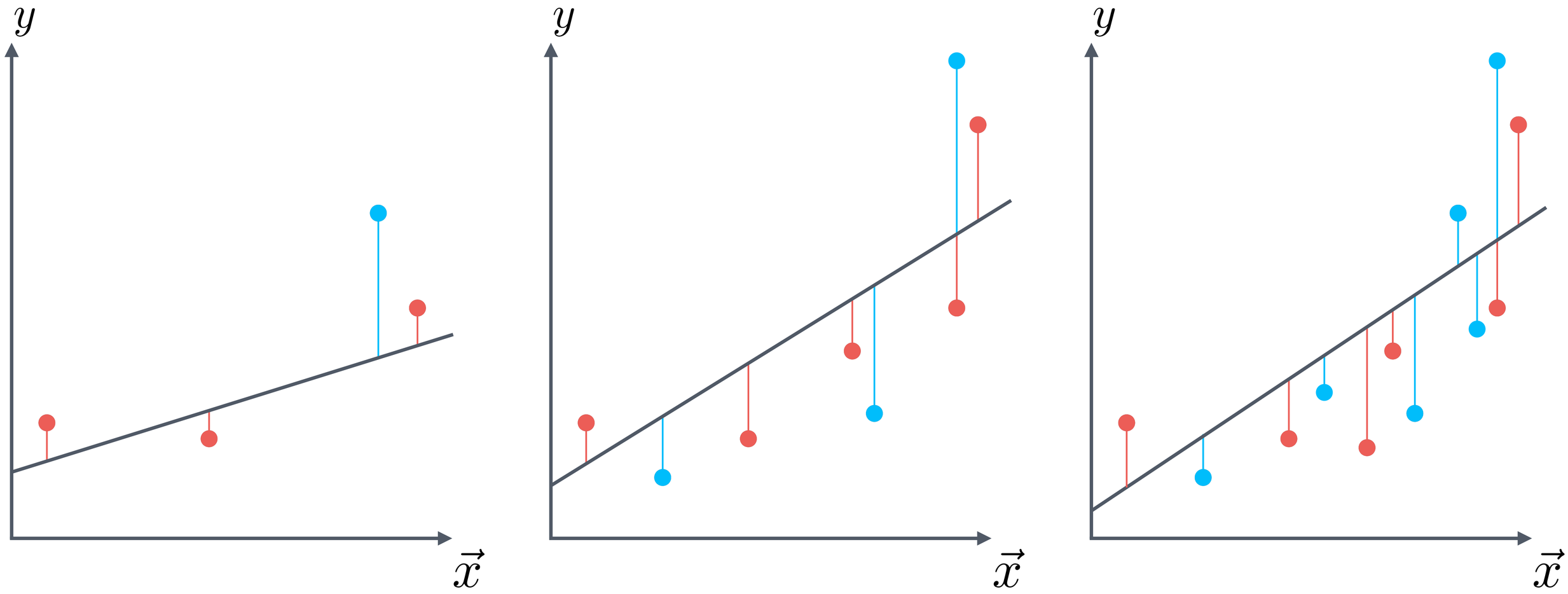


Underfitting



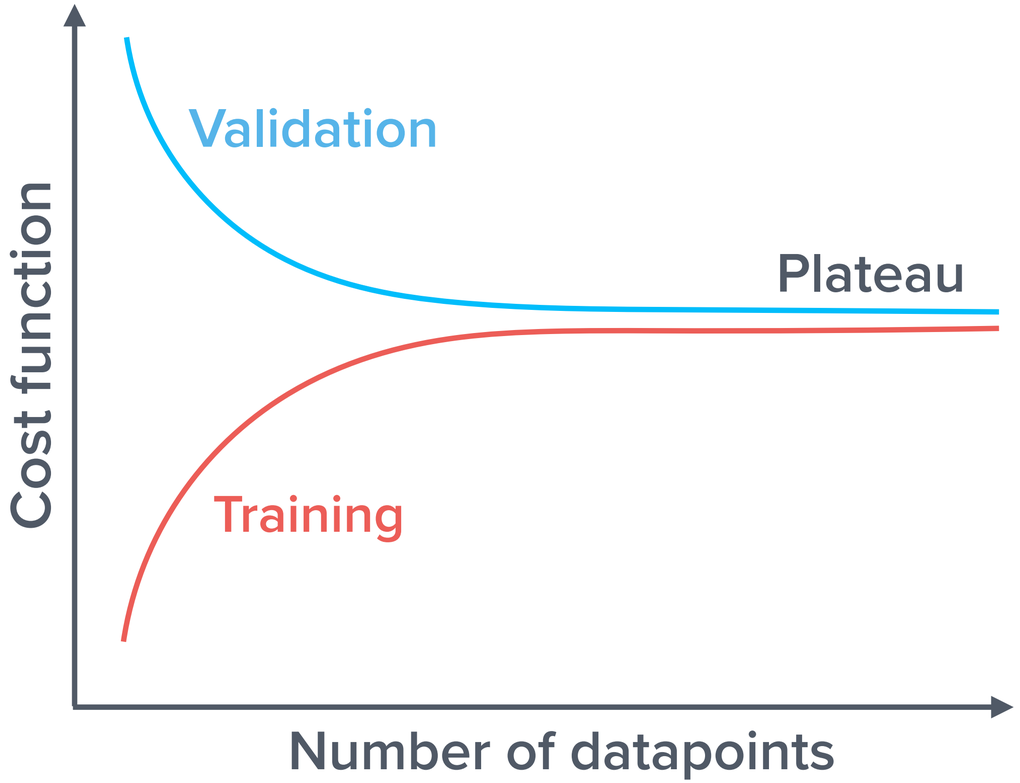
Overfitting

Underfitting: Why the plateau?

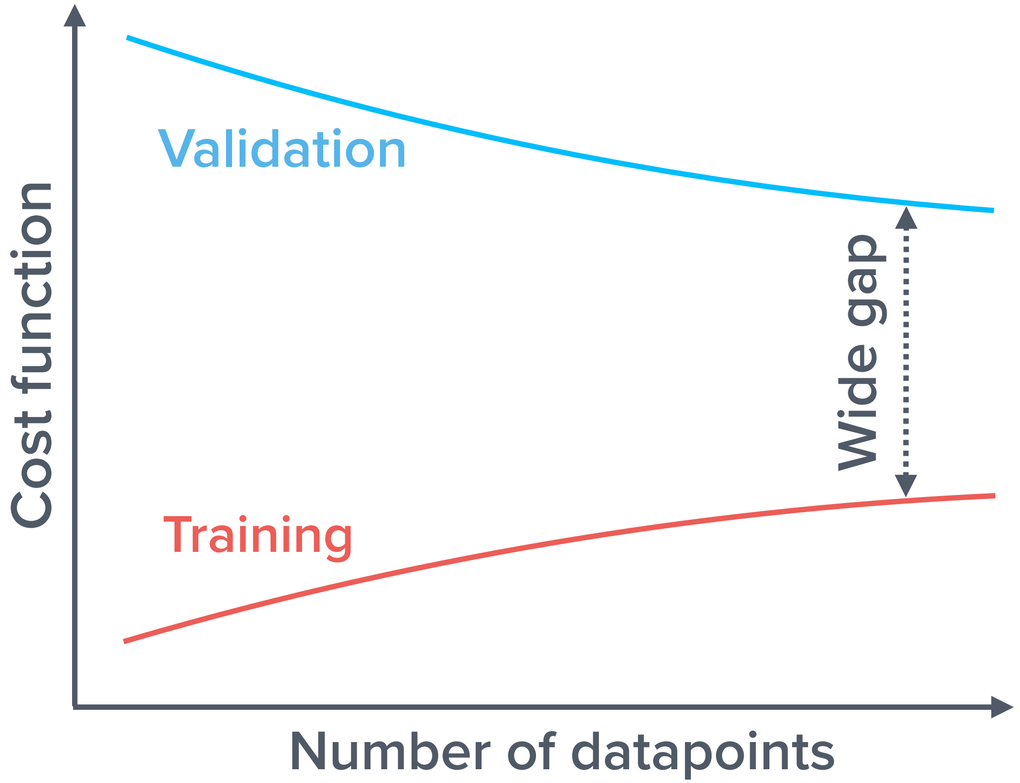


Collecting **more data** eventually generates **no improvement**.

Diagnosing underfitting vs overfitting #2

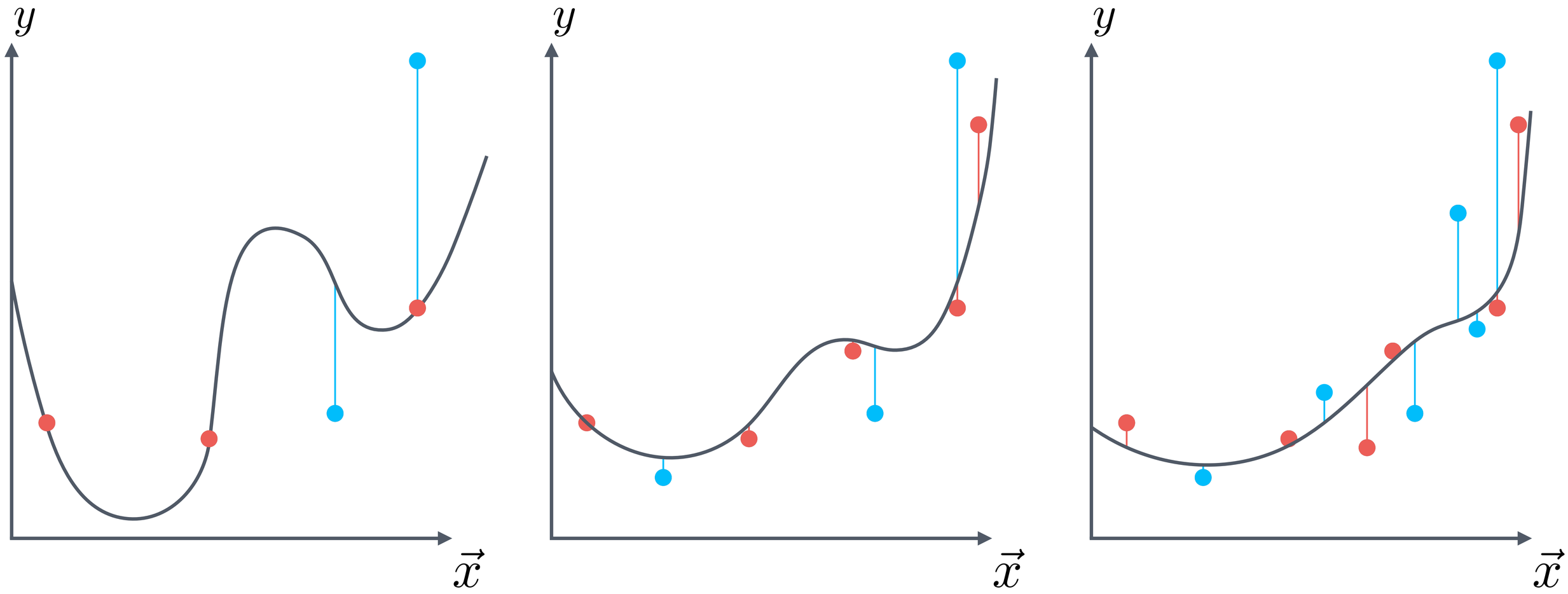


Underfitting



Overfitting

Overfitting: Why the large gap?



Collecting **more data** generates **slow improvement**.