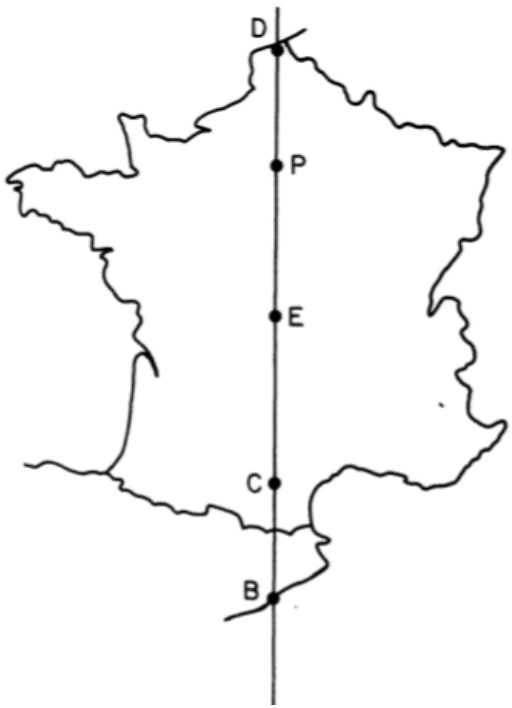
School of Electronic Engineering and Computer Science
Queen Mary University of London

ECS7020P Machine Learning
Supervised learning: Regression

Dr Jesús Requena Carrión

5 Oct 2021

How far is the equator from the north pole?



"By using this method, a sort of equilibrium is established between the **errors** which prevents the extremes from prevailing [...] [getting us closer to the] **truth**."

Adrien-Marie Legendre, 1805

Embrace the error!

Agenda

Recap

Formulation of regression problems

Basic regression models

Flexibility, interpretability and generalisation

Summary

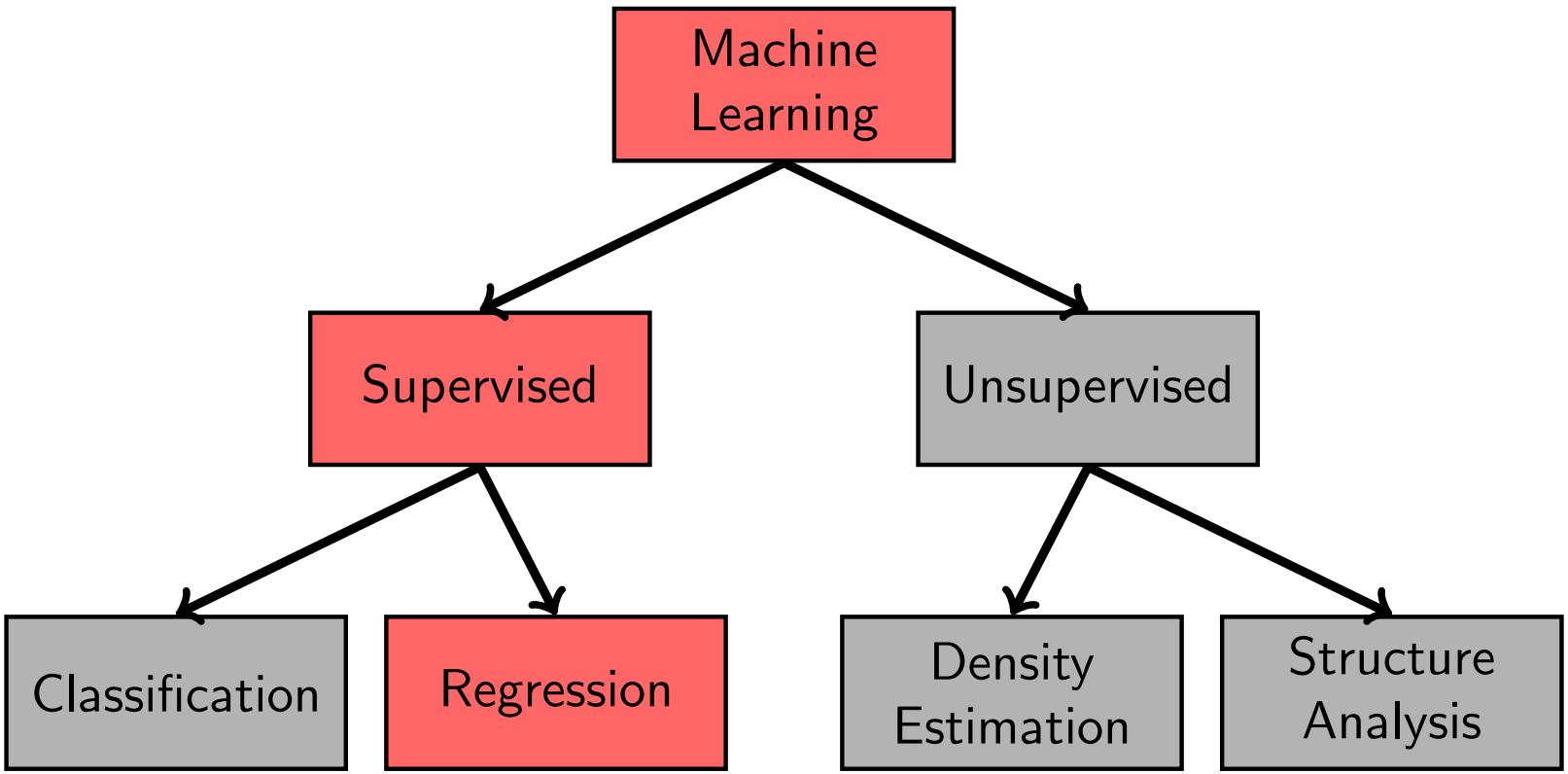
Machine learning

In machine learning we extract **knowledge** from **data**.

Knowledge is represented as a mathematical or computer **model** and data is organised as a **dataset** (a collection of **items** described by a set of **attributes**).

Machine learning distinguishes between different types of problems, techniques and models, which can be arranged into a **taxonomy**.

Machine learning taxonomy



Agenda

Recap

Formulation of regression problems

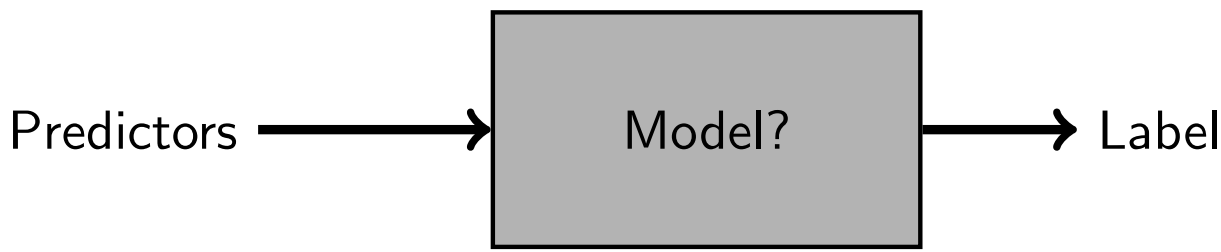
Basic regression models

Flexibility, interpretability and generalisation

Summary

Problem formulation

- Regression is a **supervised** problem: Our goal is to predict the value of one of the attributes (**label**) using the remaining attributes (**predictors**).
- The label is a **continuous variable**.
- Our job is then to **find the best model** that associates a unique label to a given set of predictors.
- We use a **dataset** consisting of **labelled samples**.



Predictors and labels

	Age	Salary
S_1	18	12000
S_2	37	68000
S_3	66	80000
S_4	25	45000
S_5	26	30000
...	...	...

In this dataset:

(a) *Age* is the predictor, *Salary* is the label

(b) *Salary* is the predictor, *Age* is the label

(c) Both options can be considered

Association and causation

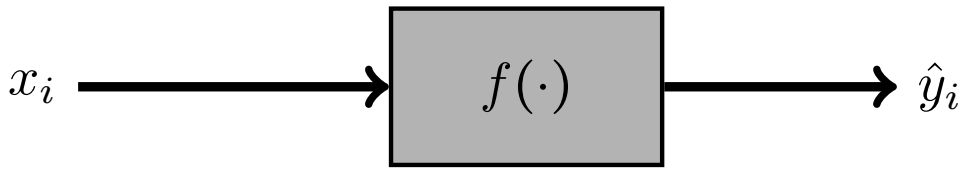
Prediction models are sometimes interpreted through a **causal lens**: the predictor is the **cause**, the label its **effect**. However this is **not correct**.

Our ability to build predictors is due to **association** between attributes, rather than **causation**. Two attributes in a dataset appear associated:

- If one causes the other (directly or indirectly).
- When both have a common cause.
- Due to the way we collect samples (sampling).

Take-home message: In machine learning we don't build causal models!

Mathematical notation



Dataset:

- N is the number of samples, i identifies each sample
- x_i is the **predictor** of sample i
- y_i is the (continuous) **true label** of sample i
- The dataset is $\{(x_i, y_i) : 1 \leq i \leq N\}$, and (x_i, y_i) is sample i

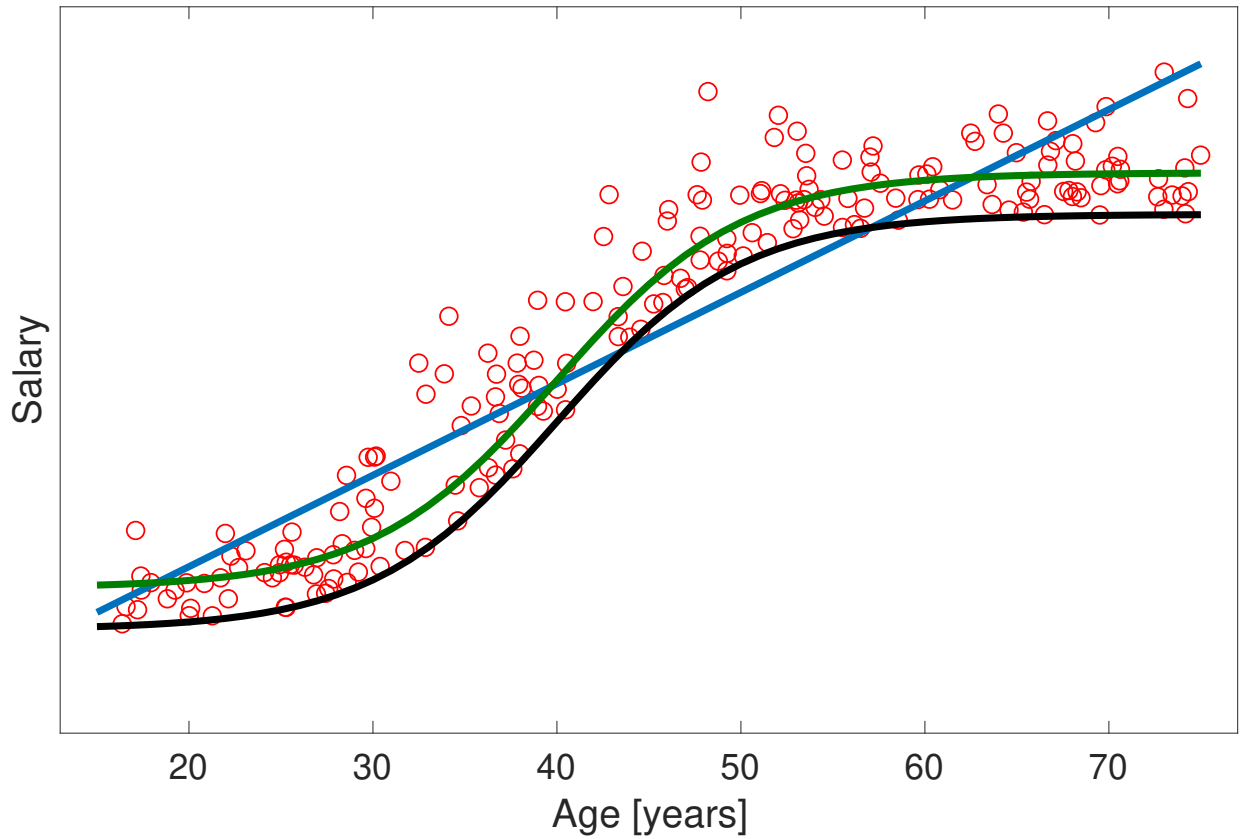
Model:

- $f(\cdot)$ denotes the model
- $\hat{y}_i = f(x_i)$ is the **predicted label** for sample i
- $e_i = y_i - \hat{y}_i$ is the **prediction error** for sample i

(*Note that we are considering* **one predictor** *here, this notation will be extended to multiple predictors when discussing multivariate models*)

Candidate solutions

Which line is the *best* mapping of age to salary?



What is a good model?

In order for us to find the **best** model we need a notion of **model quality**.

One popular quality metric in regression problems is the **mean squared error** (MSE), which corresponds to the expected squared error of the prediction of a model during deployment.

If we are given a dataset consisting of N samples and a model $f(\cdot)$, we can **estimate** its MSE as follows:

$$\begin{aligned} E_{MSE} &= \frac{1}{N}\sum_{i=1}^{N} e_i^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} (y_i - f(x_i))^2 \end{aligned}$$

MSE: Example

A zero-error model?

Given a dataset, is it possible to find a model such that $\hat{y}_i = y_i$ for every instance i in the dataset, i.e. a model whose **error is zero**, $E_{MSE} = 0$?

(a) **Never**, there will always be a non-zero error

(b) It is **never guaranteed**, but might be possible for some datasets

(c) **Always**, there will always be a model complex enough that achieves this

The nature of the error

When considering a regression problem we need to be aware that:

- The chosen **predictors might not include all the factors** that determine the label.
- The chosen **model might not be able to represent accurately** the true relationship between response and predictor (the pattern).
- **Random mechanisms** (noise) might be present.

Mathematically, we represent this discrepancy as

$$\begin{aligned} y &= \hat{y} + e \\ &= f(x) + e \end{aligned}$$

There will always be some discrepancy (error e) between the true label y and our model prediction $f(x)$. **Embrace the error!**

Regression as an optimisation problem

Given a dataset $\{(x_i, y_i) : 1 \leq i \leq N\}$, every candidate model f has its own E_{MSE}. Our goal is to find the **model with the lowest** E_{MSE}:

$$f_{best}(x) = \arg\min_{f} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2$$

The question is, how do we find such model? Finding such a model is an **optimisation problem**.

Notice that we are looking for the model that minimises E_{MSE} **on the dataset**, however this model might not be the **minimum MSE** (MMSE) solution, i.e. the best model during **deployment**!

Agenda

Recap

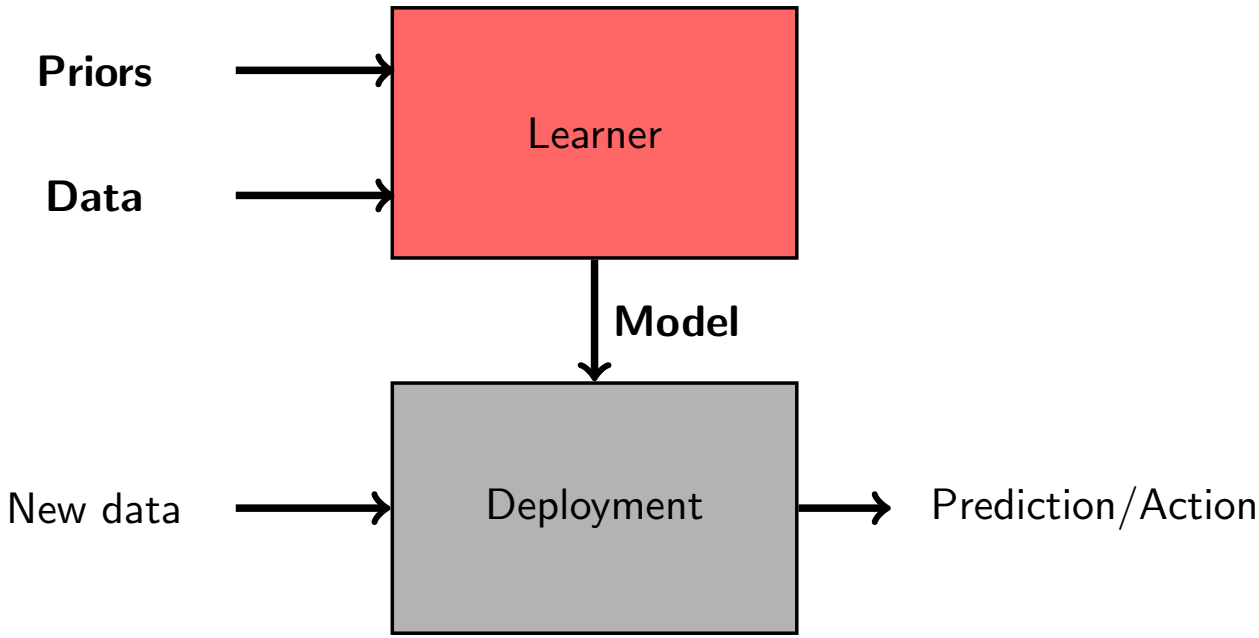
Formulation of regression problems

Basic regression models

Flexibility, interpretability and generalisation

Summary

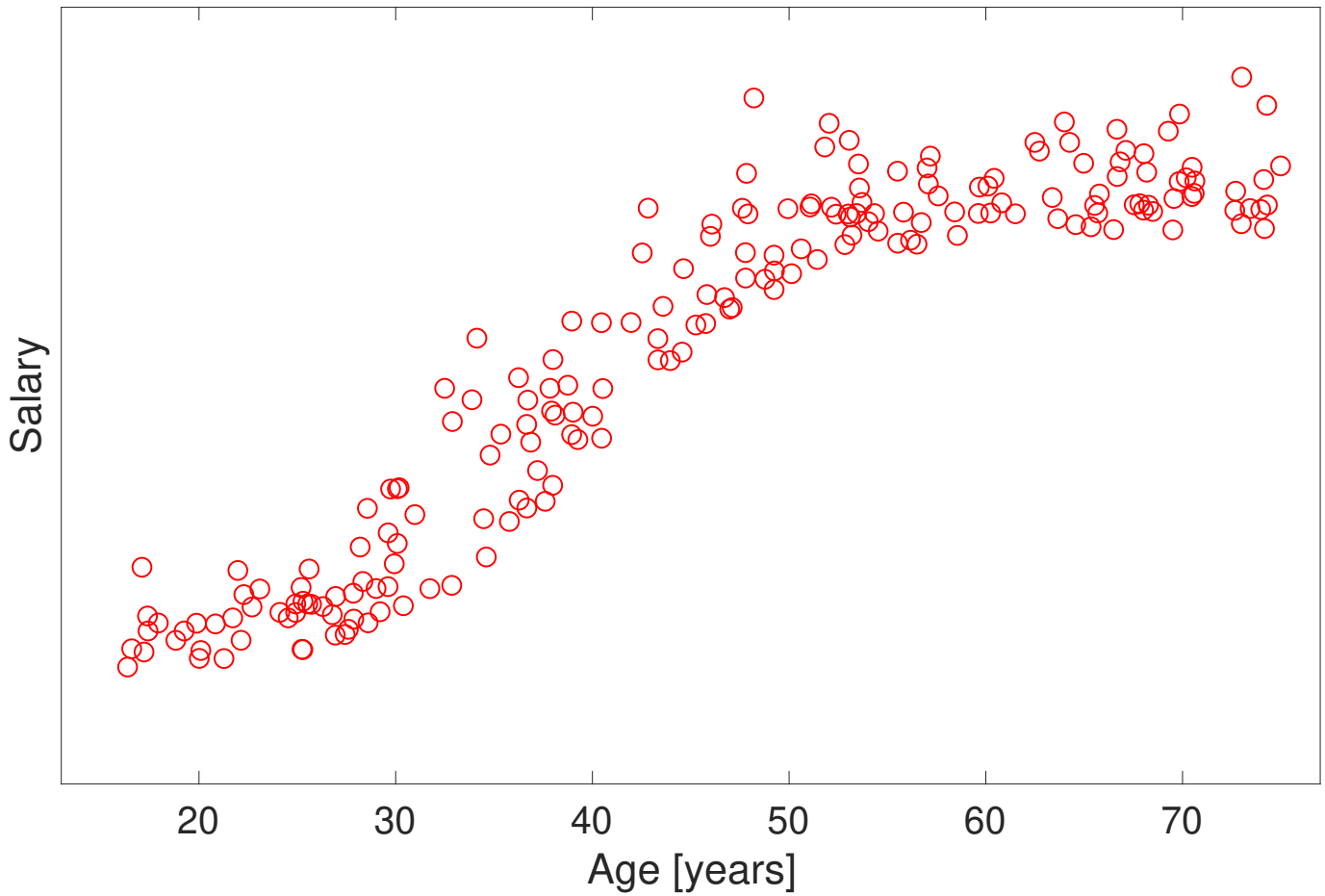
Our regression learner



- **Priors**: Type of model (linear, polynomial, etc).
- **Data**: Labelled samples, predictors and true (continuous) label.
- **Model**: Predicts a label based on the predictors.

Simple regression

Simple regression considers **one predictor** x and one label y.



Simple linear regression

In simple **linear** regression, **models** are defined by the mathematical expression

$$f(x) = w_0 + w_1 x$$

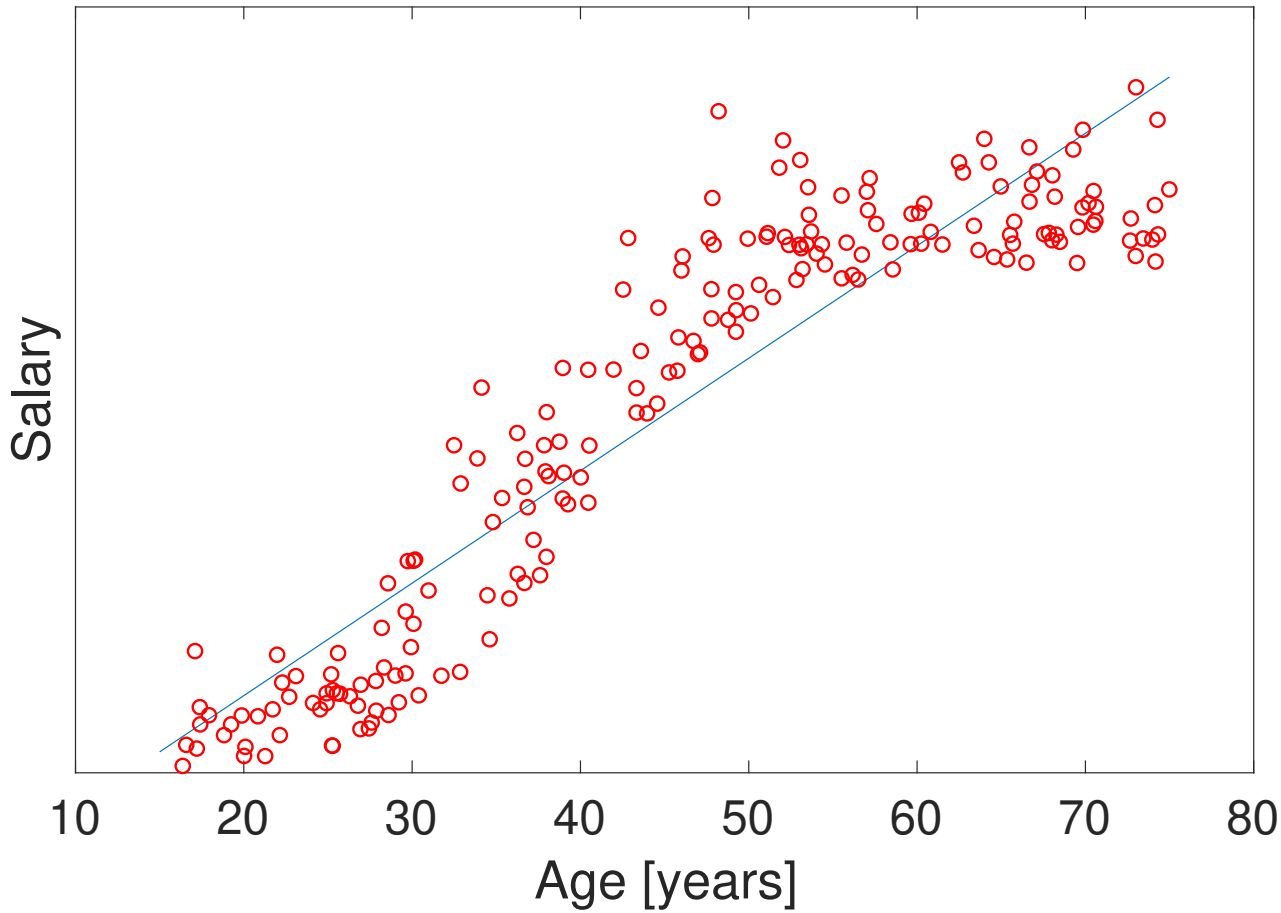
Hence, the predicted label $\hat{y}_i$ can be expressed as

$$\hat{y}_i = f(x_i) = w_0 + w_1 x_i$$

A linear model has therefore **two parameters** w_0 (intercept) and w_1 (slope), which need to be **tuned** to achieve the highest quality.

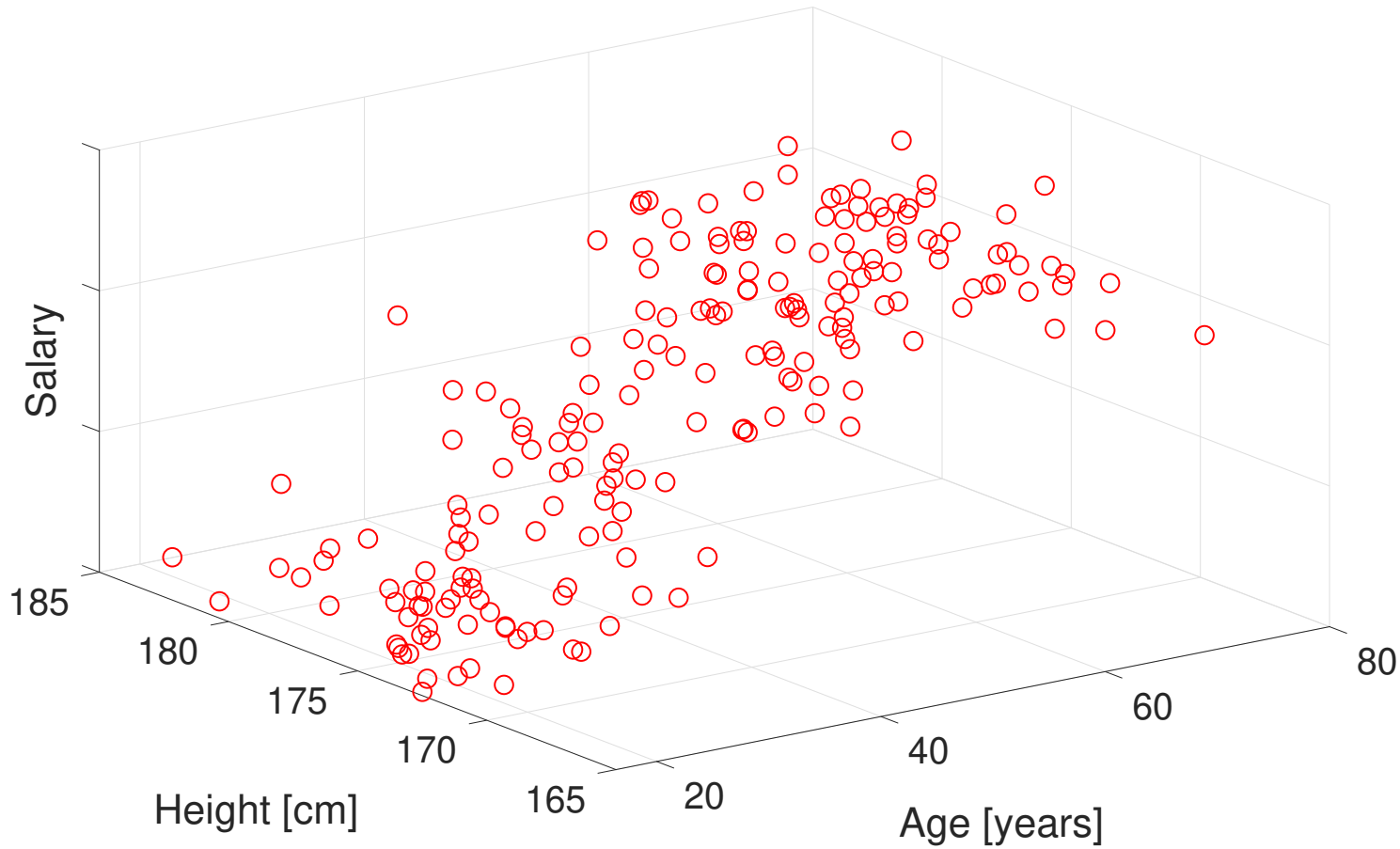
In machine learning, we use a **dataset** to tune the parameters. We say that we **train the model** or **fit the model** to the **training dataset**.

Linear solution: Example



Multiple linear regression

In multiple regression there are **two or more predictors**.



Multiple regression: Notation

Using vector notation, predictors can be expressed as the **extended vector**:

$$\boldsymbol{x}_i = [1, x_{i,1}, x_{i,2}, \ldots, x_{i,K}]^T,$$

where $x_{i,k}$ denotes the k-th predictor of the i-th sample and K is the number of predictors. The constant 1 is prepended for convenience.

Multiple regression can then be expressed as

$$\hat{y}_i = f(\boldsymbol{x}_i)$$

Good news: the notation developed for simple regression can be easily translated to the multivariate scenario, no extra efforts required.

Multiple linear regression: Formulation

Multiple **linear** regression models can be expressed as:

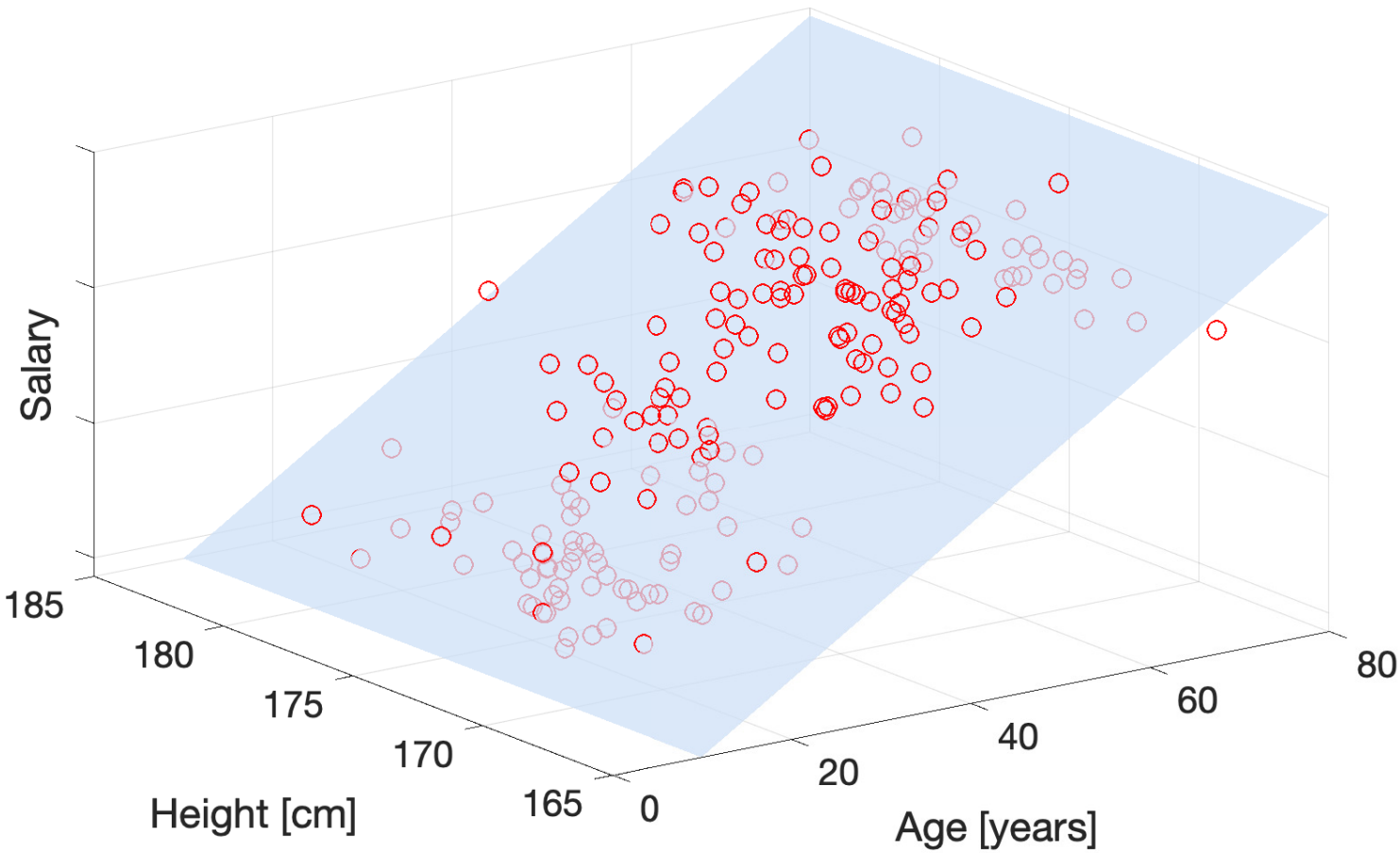
$$f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i = w_0 + w_1 x_{i,1} + \cdots + w_K x_{i,K}$$

where $\boldsymbol{w} = [w_0, w_1, \ldots, w_K]^T$ is the model's parameter vector.

Note that we can use the same vector notation for simple linear regression models, by defining $\boldsymbol{w} = [w_0, w_1]^T$ and $\boldsymbol{x}_i = [1, x_i]^T$.

Multiple linear regression: Solution visualisation

Multiple linear regression models are planes (or hyperplanes).



Multiple regression: More notation

In multiple linear regression, the **training dataset** can be represented by the **design matrix** $\mathbf{X}$:

$$\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,K} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,K} \end{bmatrix}$$

and the **label vector** $\boldsymbol{y}$:

$$\boldsymbol{y} = [y_1, \ldots, y_N]^T = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

The least squares solution

It can be shown that the **linear model** that minimises the metric E_{MSE} on a **training dataset** defined by a design matrix $\mathbf{X}$ and a label vector $\mathbf{y}$, has the parameter vector:

$$\boldsymbol{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{y}$$

This is an **exact** or **analytical solution** and is known as the **least squares** solution. It is valid for simple and multiple linear regression.

Note that the inverse matrix $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ *exists when all the columns in* $\mathbf{X}$ *are independent.*

Multiple linear regression: Example

Use vector notation to represent a multiple linear regression model where the predictors are *age* and *height* and the label is *salary*.

Multiple linear regression: Example

Consider a dataset consisting of 4 samples described by three attributes, namely age, height and salary:

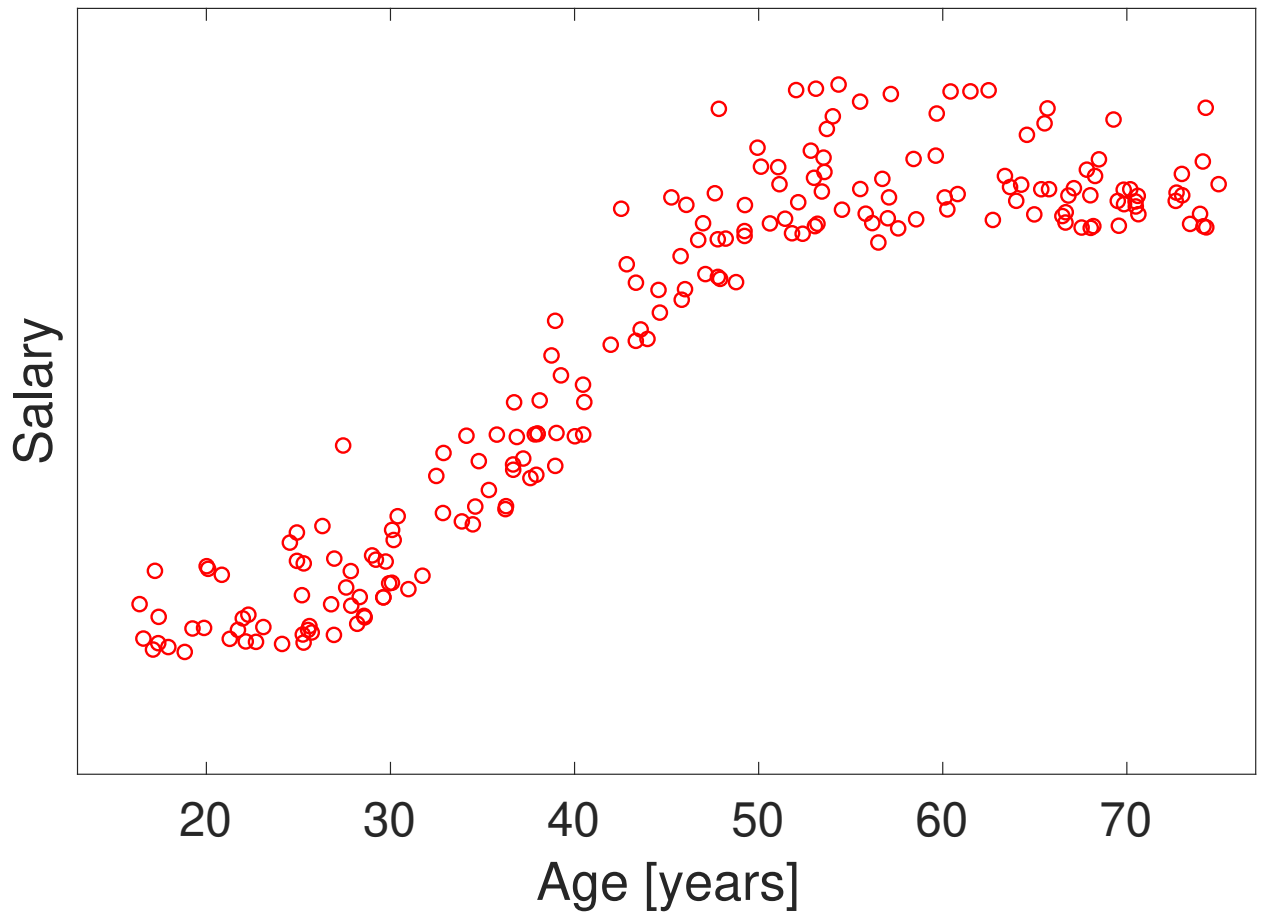
	Age [Years]	Height [cm]	Salary [GBP]
S_1	18	175	12000
S_2	37	180	68000
S_3	66	158	80000
S_4	25	168	45000

We decide to build a linear model that maps age and height to salary.

1. Use vector notation to represent the resulting linear regression model.
2. Obtain the design matrix $\mathbf{X}$ and response vector $\boldsymbol{y}$.

Beyond linearity

Sketch the model that you would choose for the Salary Vs Age dataset and try to find a suitable mathematical expression.



Simple polynomial regression

The general form of a polynomial regression model is:

$$f(x_i) = w_0 + w_1 x_i + w_2 x_i^2 + \cdots + w_D x_i^D$$

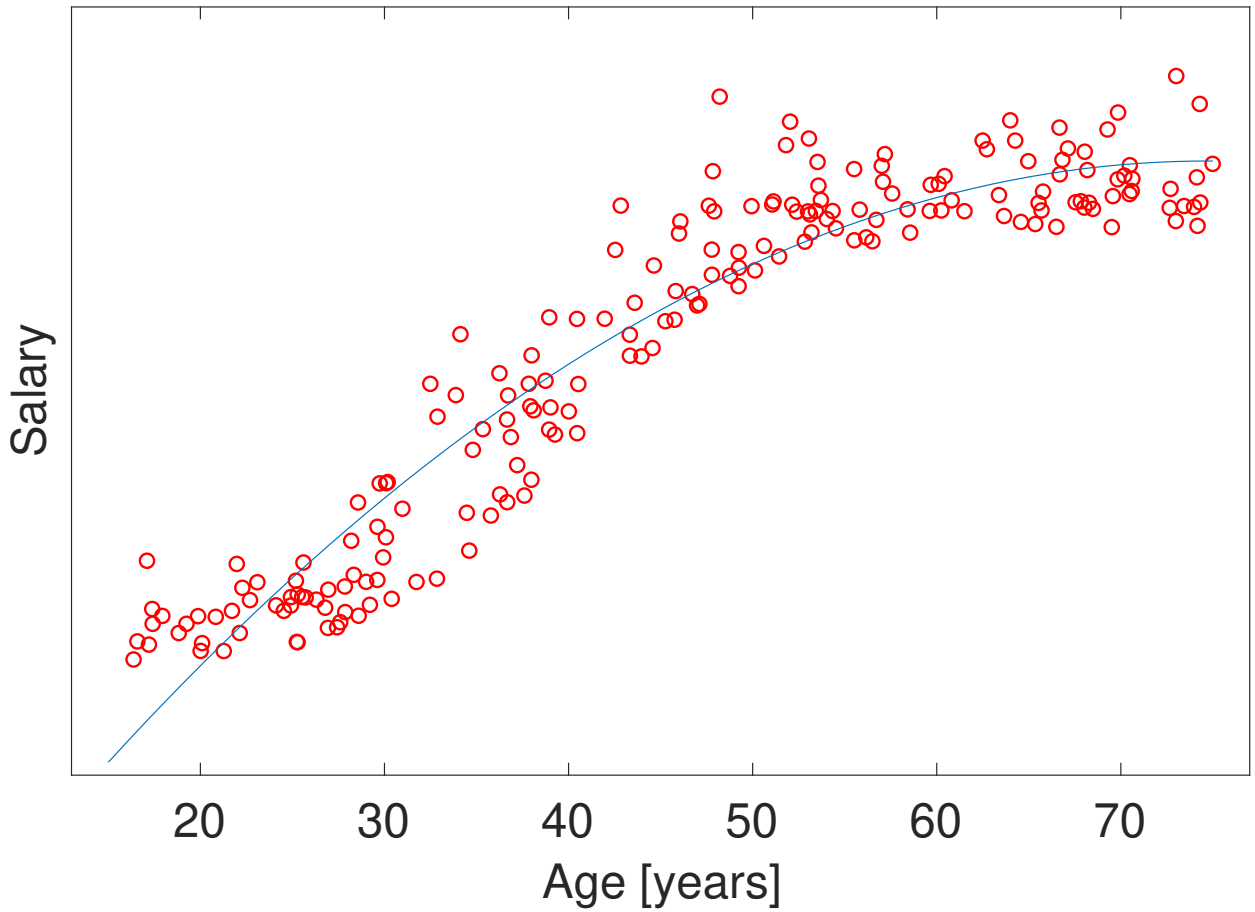
where D is the degree of the polynomial.

By treating the powers of the predictor x as predictors themselves, simple polynomial models can be expressed as multiple linear models:

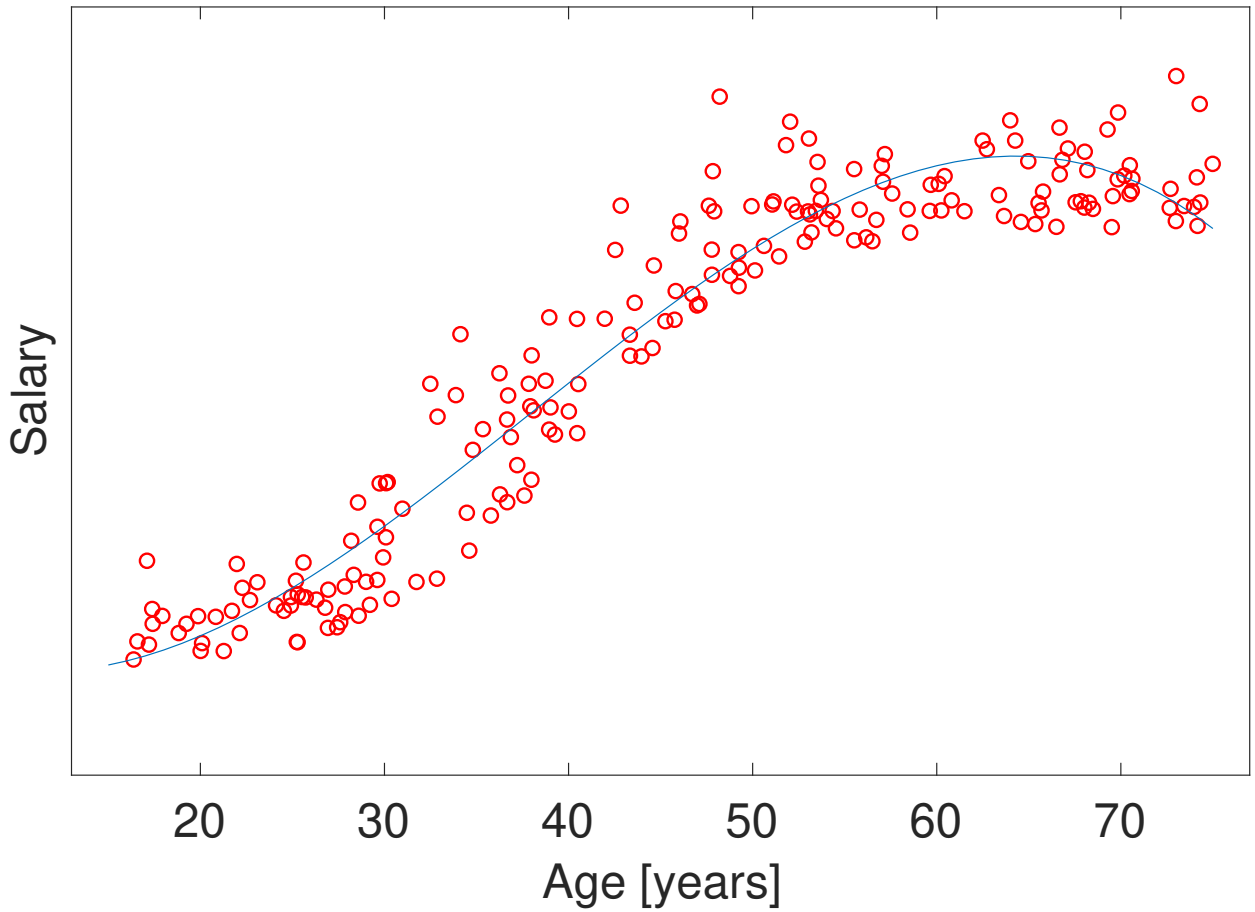
$$f(x_i) = w_0 + w_1 x_i + w_2 x_i^2 + w_3 x_i^3 = \boldsymbol{w}^T \boldsymbol{\phi}_i$$

where $\boldsymbol{\phi}_i = [1, x_i, x_i^2, x_i^3]^T$. Therefore, there is an exact **least squares solution for simple polynomial regression**.

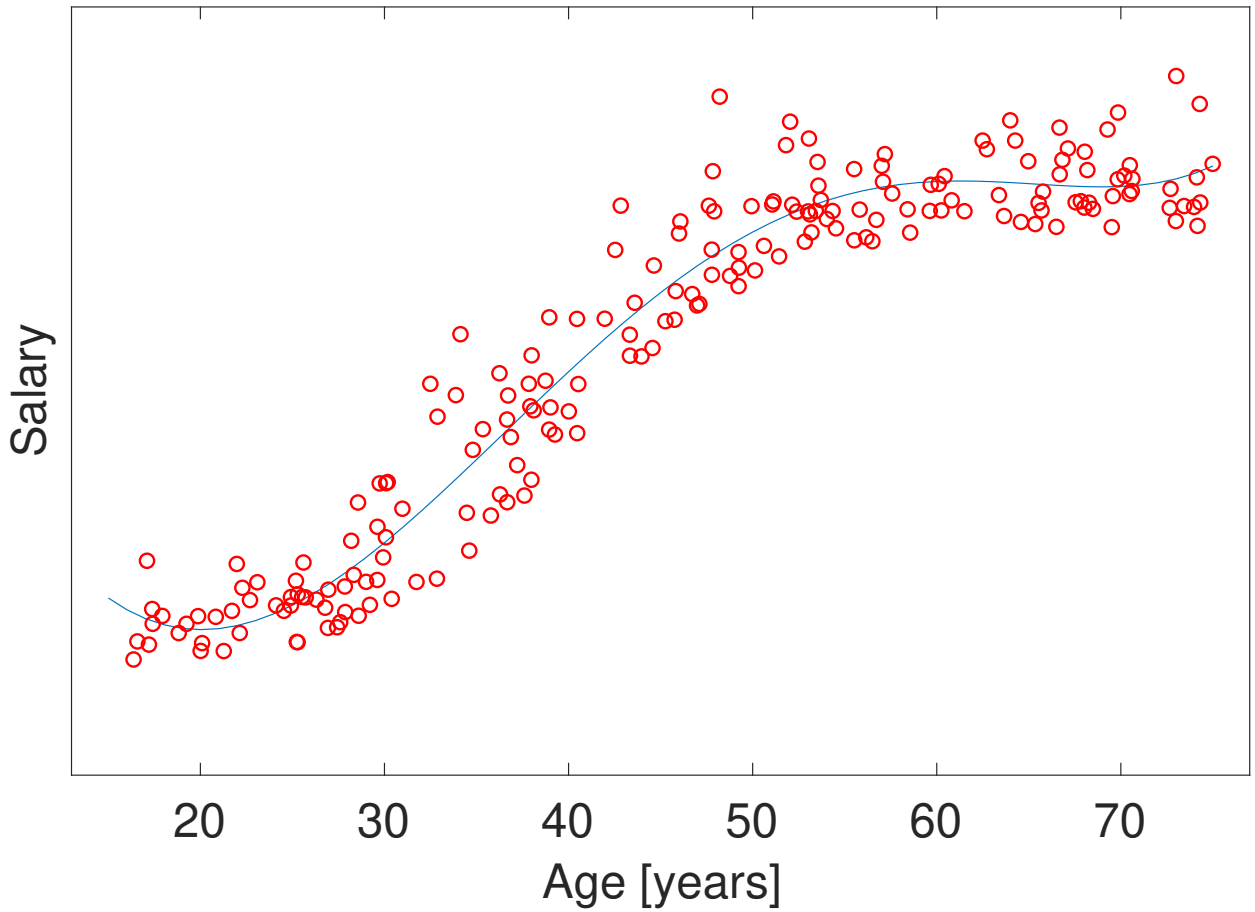
Quadratic solution



Cubic solution



5-power solution



Other models for regression

Linear and polynomial models are not the only options available. Other families of models that can be used include:

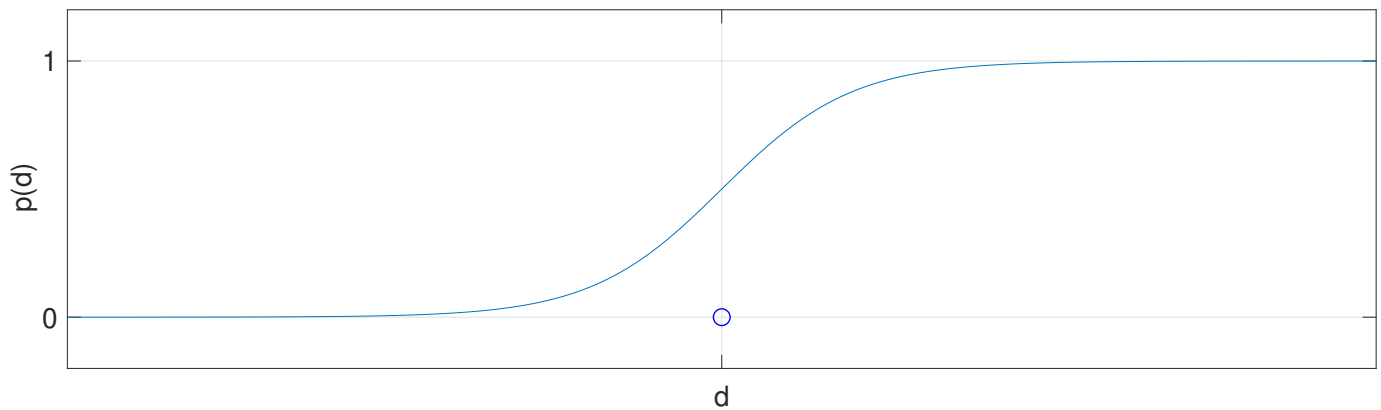
- Exponential
- Sinusoids
- Radial basis functions
- Splines
- And many more!

The mathematical formulation is identical and only the expression for $f(\cdot)$ changes.

Logistic regression

In some problems, the label represents a **proportion** or a **probability**, i.e. a quantity between 0 and 1. Moreover, this quantity might increase as the predictor increases. In such cases, the logistic function can be useful.

The logistic function $p(d)$ is defined as $p(d) = \frac{e^d}{1+e^d} = \frac{1}{1+e^{-d}}$. By using $d = w_0 + w_1 x$, we can translate the logistic function and change its slope.



Other quality metrics

In addition to the MSE, we can consider other quality metrics:

- **Root mean squared error**. Measures the sample standard deviation of the prediction error.

$$E_{RMSE} = \sqrt{\frac{1}{N}\sum e_i^2}$$

- **Mean absolute error**. Measures the average of the absolute prediction error.

$$E_{MAE} = \frac{1}{N}\sum |e_i|$$

- **R-squared**. Measures the proportion of the variance in the response that is predictable from the predictors.

$$E_R = 1 - \frac{\sum e_i^2}{\sum (y_i - \bar{y})^2}, \text{ where } \bar{y} = \frac{1}{N}\sum y_i$$

Agenda

Flexibility

Models allow us to generate multiple shapes by tuning their parameters. We talk about the **degrees of freedom** or the **complexity** of a model to describe its ability to generate different shapes, i.e. its **flexibility**.

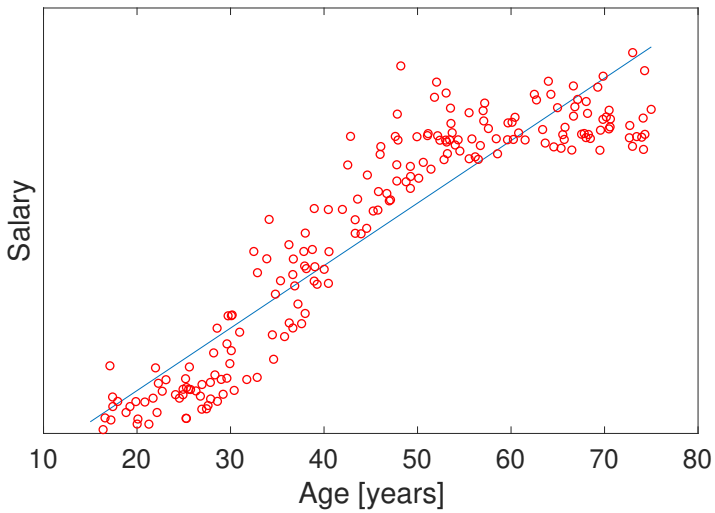
The degrees of freedom of a model are in general related to the number of parameters of the model:

- A linear model $y = w_0 + w_1 x$ has two parameters and is inflexible, as it can only generate straight lines.
- A cubic model $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$ has 4 parameters and is more flexible than a linear one.

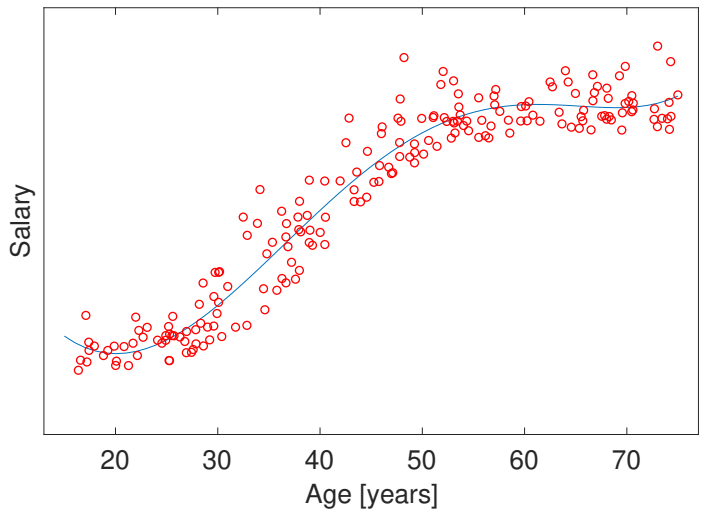
The flexibility of a model is related to its **interpretability** and **accuracy** and there is a trade-off between the two.

Interpretability

Model interpretability is crucial for us, as humans, to understand in a qualitative manner how a predictor is mapped to a label. Inflexible models produce solutions that are usually simpler and easier to interpret.



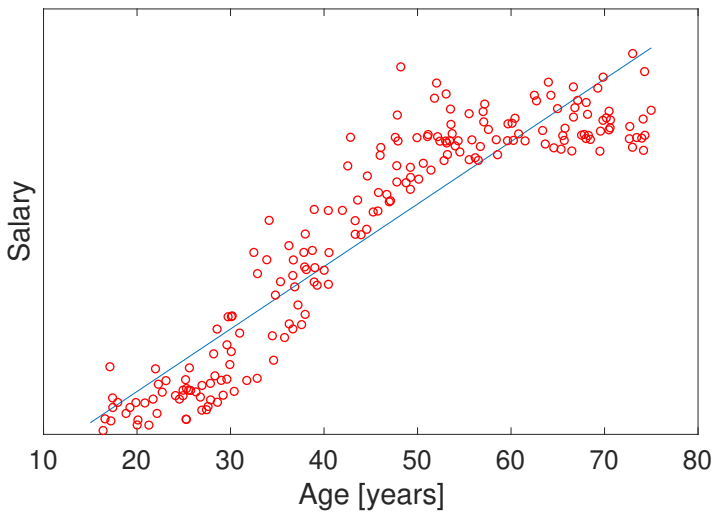
According to this linear model, the older you get, the more money you make



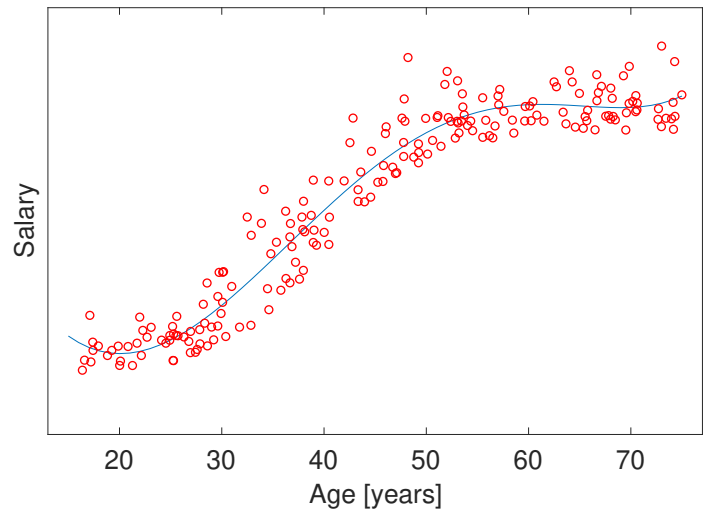
According to this polynomial model, our salary remains the same as teenagers, then increases between our 20s and 50s, then...

Accuracy

The accuracy of a model is also related to its flexibility. During training, the error produced by flexible models is in general lower.



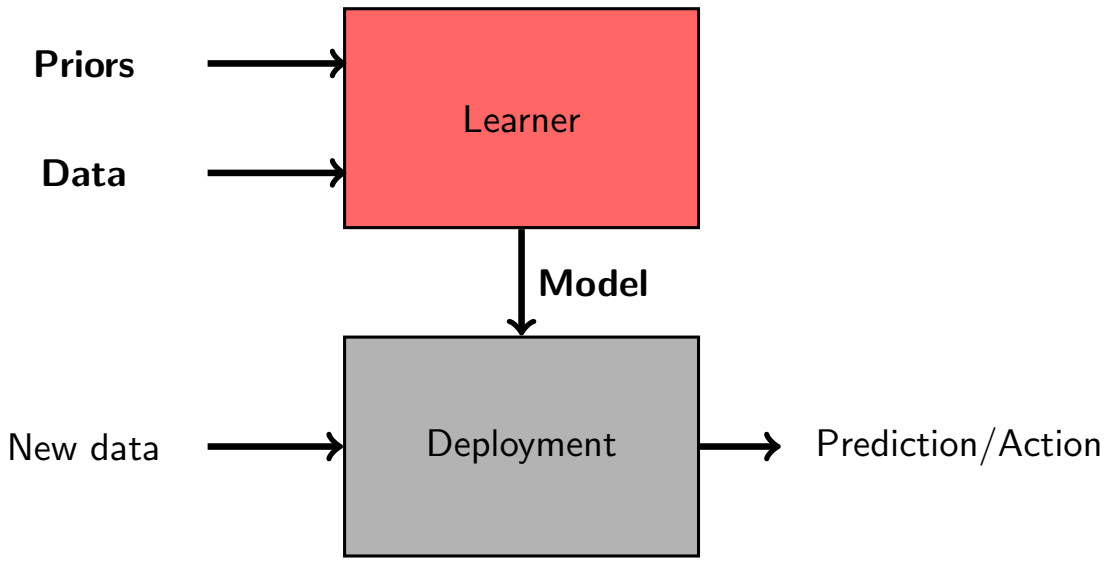
The **training error** of the best linear model is $E_{MSE} = 0.0983$



The **training error** of the best polynomial model is $E_{MSE} = 0.0379$

Generalisation

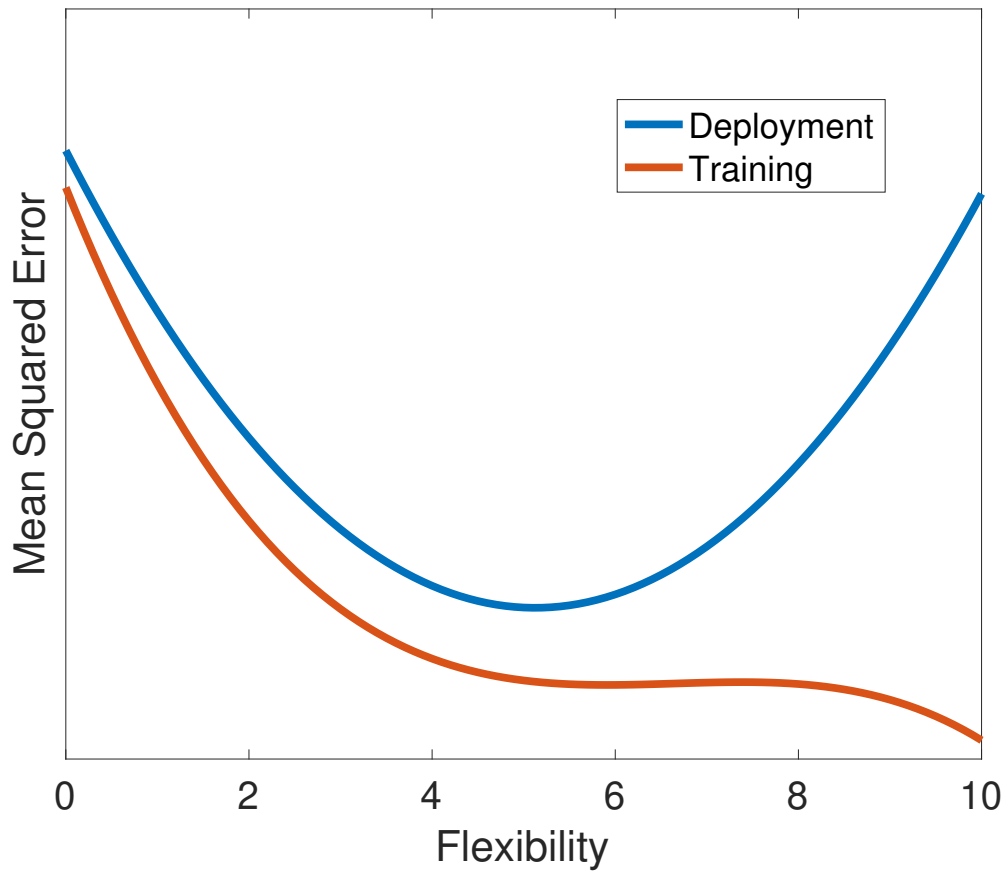
We have considered the **training MSE**, i.e. the quality of regression models on the **training dataset**.



Will our model work well during deployment, when presented with new data? **Generalisation** is the ability of our model to successfully translate what we was learnt during the learning stage to deployment.

Generalisation

In this figure, the red curve represents the **training MSE** of different models of increasing complexity, whereas the blue curve represents the **deployment MSE** for the same models. What's happening?

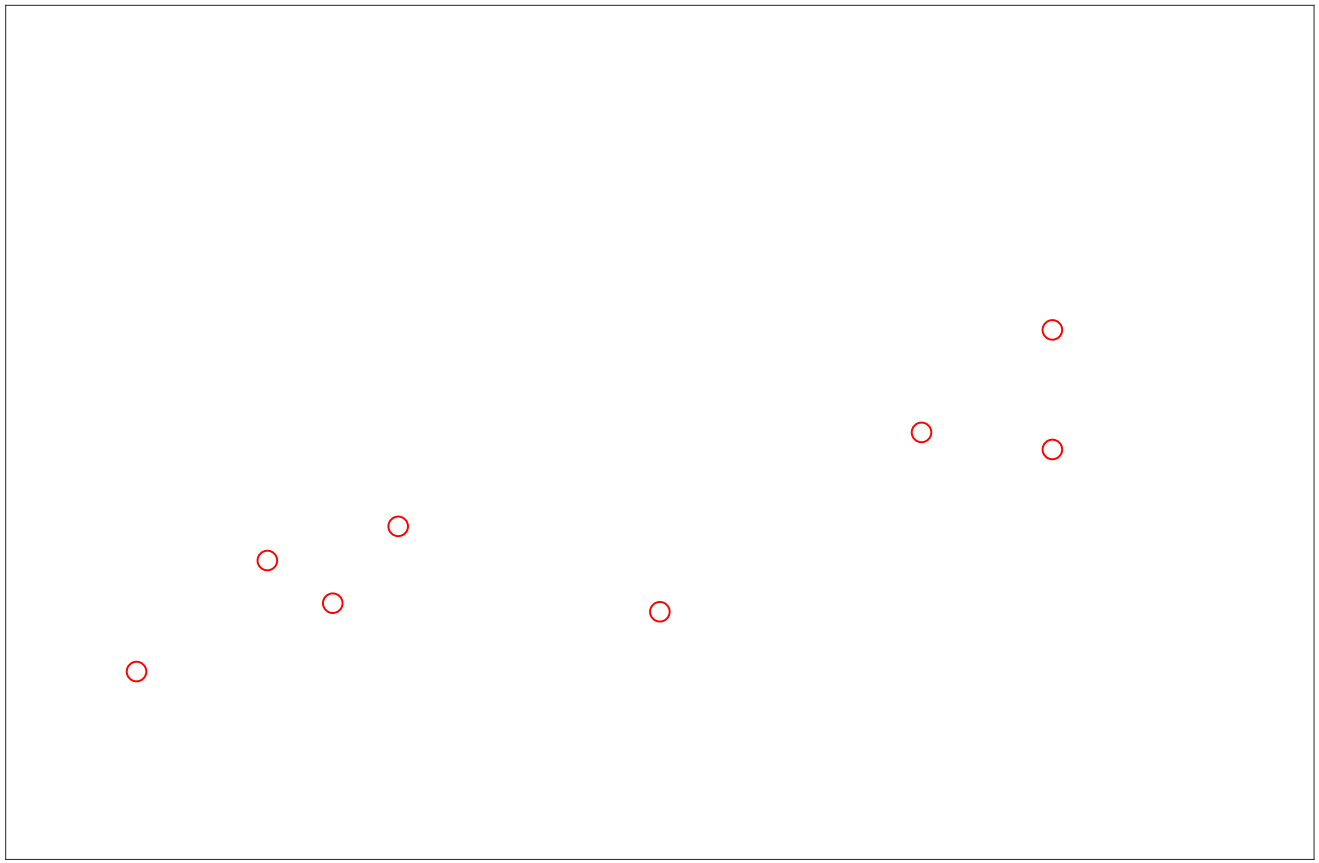


Underfitting and overfitting

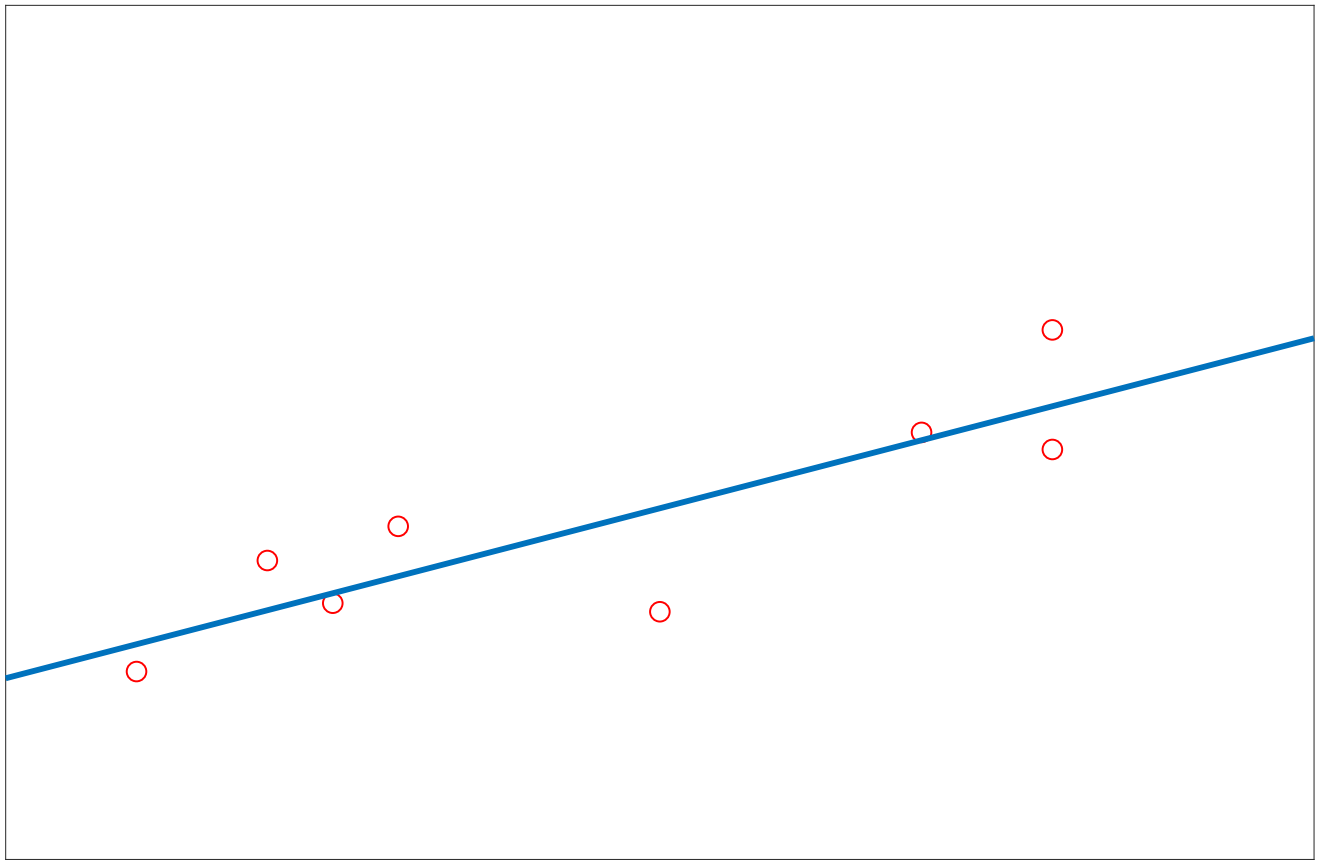
By comparing the performance of models during training and deployment, we can observe three different behaviours:

- **Underfitting**: Large training and deployment errors are produced. The model is unable to capture the **underlying pattern**. Rigid models lead to underfitting.
- **Overfitting**: Small errors are produced during training, large errors during deployment. The model is memorising **irrelevant details**. Too complex models and not enough data lead to overfitting.
- **Just right**: Low training and deployment errors. The model is capable of reproducing the **underlying pattern** and ignores **irrelevant details**.

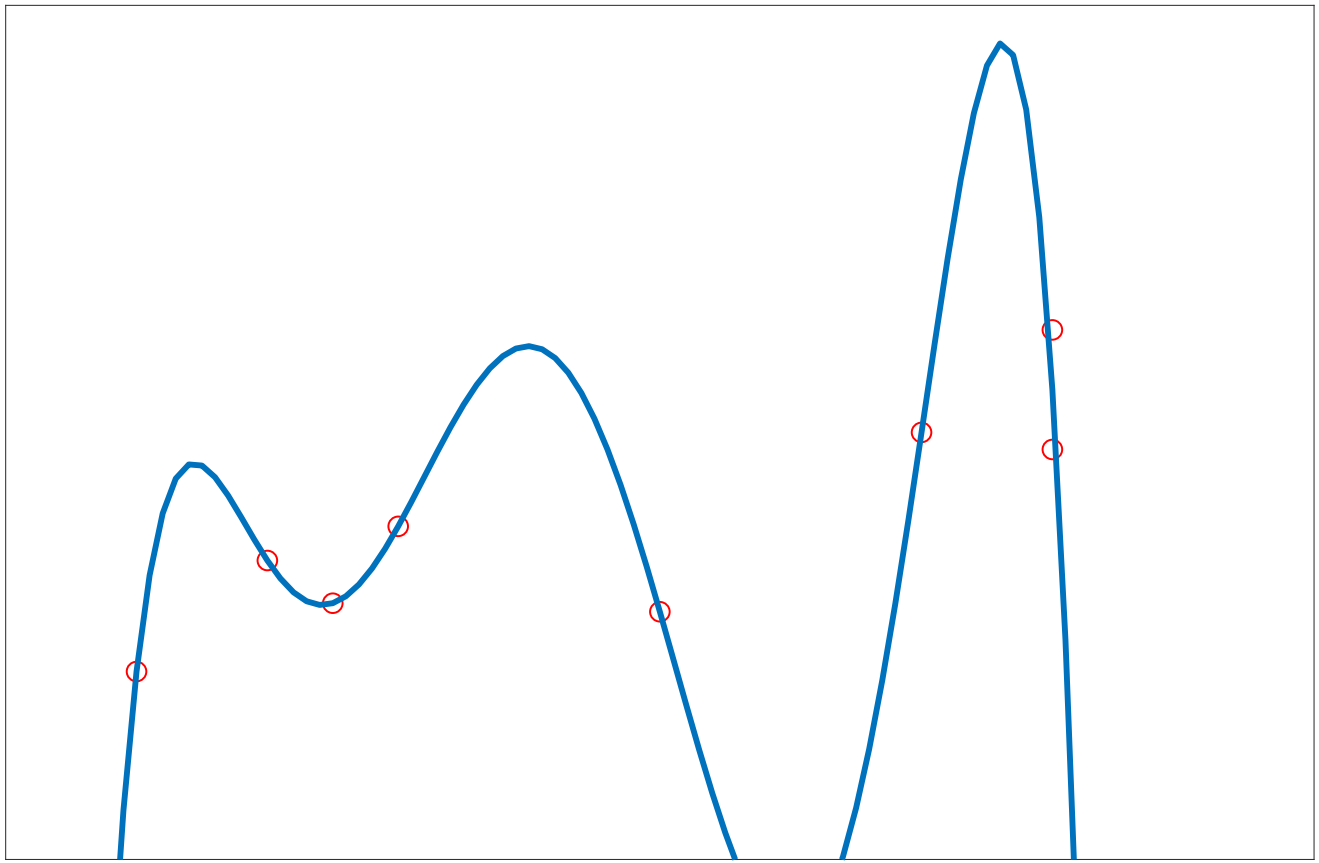
Underfitting and overfitting



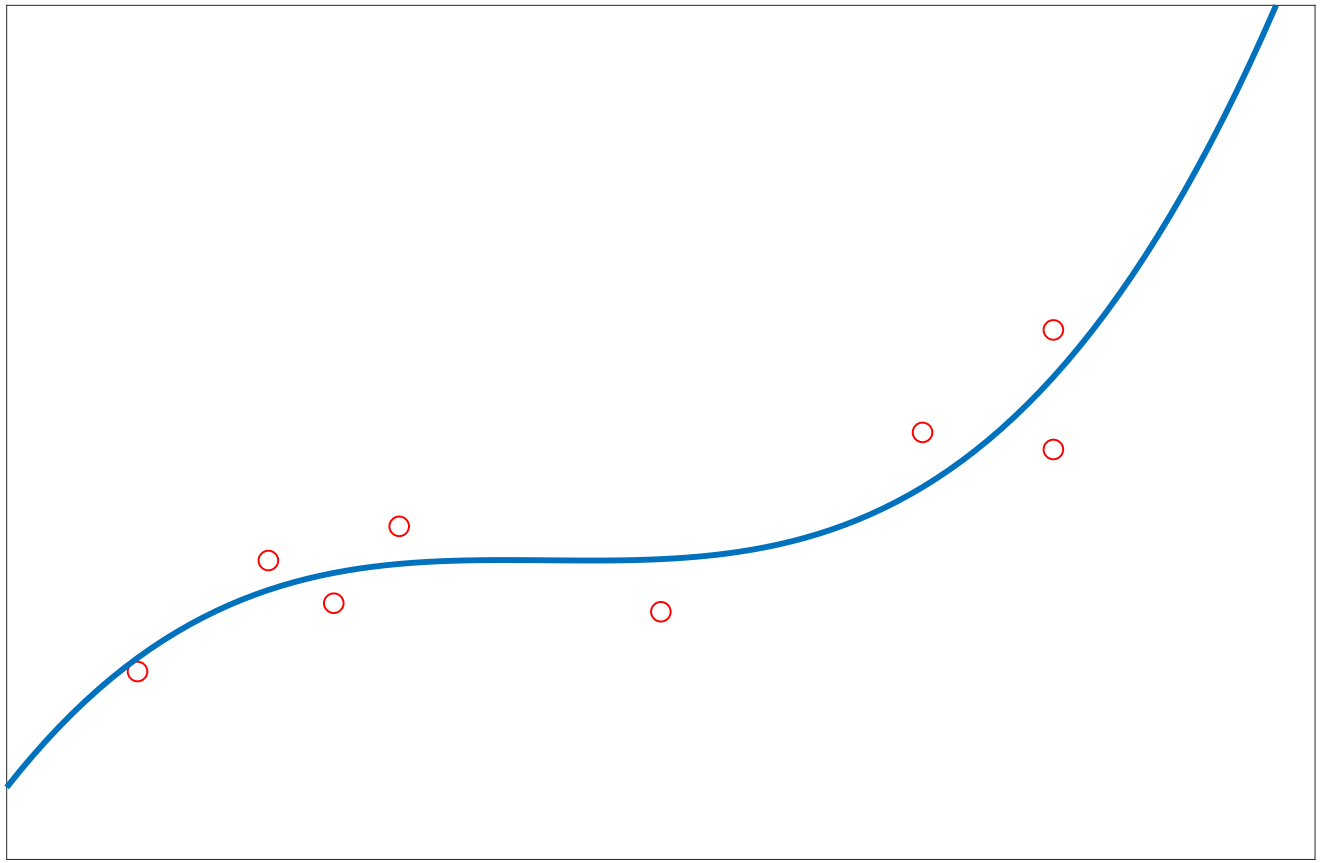
Underfitting



Overfitting

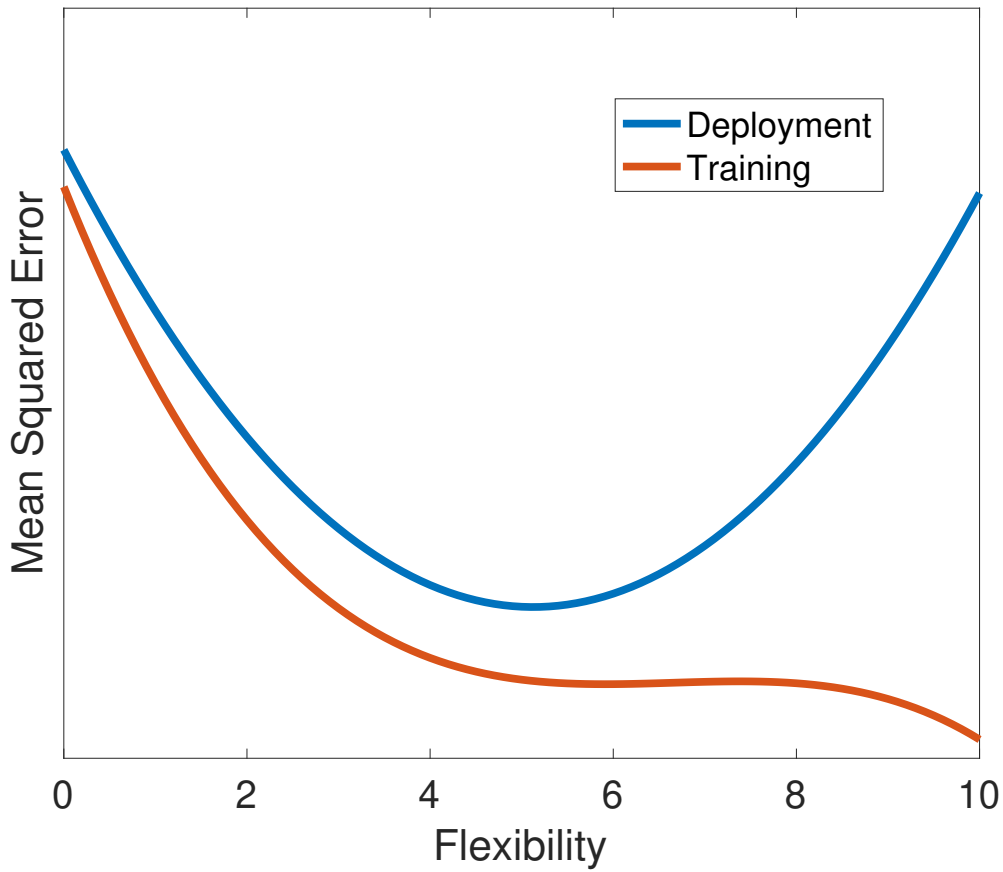


Just right



Underfitting and overfitting

Remember this: Generalisation can only be assessed by comparing training and deployment performance, not by just looking at how each model fits the training data.



Agenda

Recap

Formulation of regression problems

Basic regression models

Flexibility, interpretability and generalisation

Summary

Regression: Basic methodology

- Regression is a family of problems in machine learning, where we set out to find a model that **predicts a continuous label**.
- To build a model we use:
  - A **training dataset**,
  - a tunable **model**,
  - a **quality metric** and
  - an **optimisation** procedure.
- The final quality of a model has to be assessed during **deployment**.

Model generalisation

- Models have different degrees of **flexibility**. Complex models are flexible, simple models are rigid.
- A model **generalises** well when it can deal successfully with samples that it hasn't been exposed to during training.
- Three terms describe the ability of models to generalise:
  - **Underfitting**: unable to describe the underlying pattern
  - **Overfitting**: memorisation of irrelevant details
  - **Just right**: reflects underlying pattern and ignores irrelevant details

Final historical note

Wondering where the term *regression* comes from?

In the 19th century, Galton noticed that children of tall people tend to be taller than average – but not as tall as their parents. Galton called this *reversion* and later *regression towards mediocrity*.

This observation is nowadays called *regression to the mean*. You can read more about this curious fallacy in Kanehman's *Thinking, Fast and Slow*.