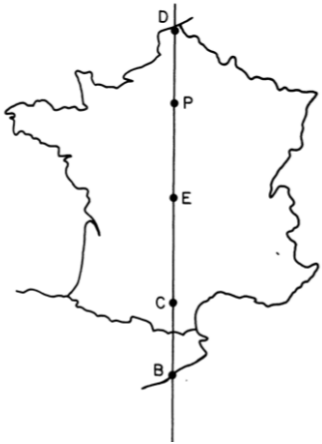
Queen Mary School Hainan
Queen Mary University of London

QHM5703 Machine Learning
Supervised learning: Regression

Dr Nikesh Bajaj

Week9 - (5/6) Nov 2025

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

Machine learning

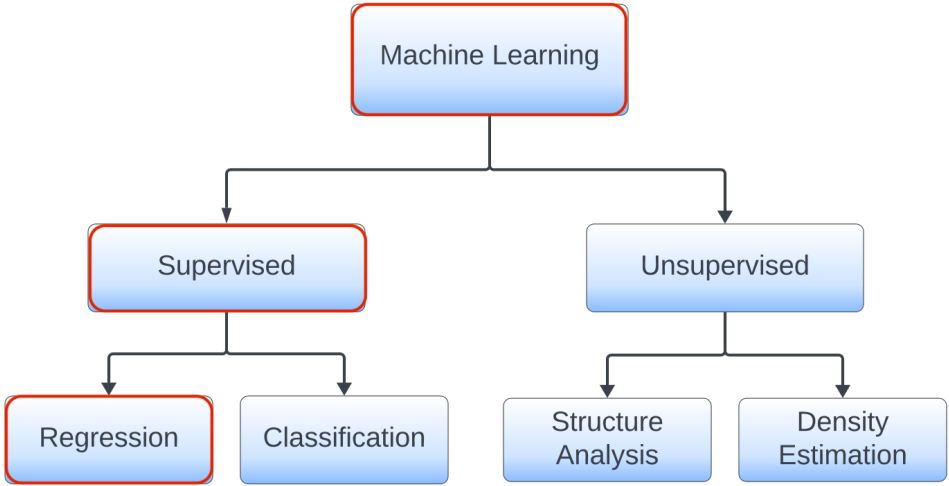
There are two main ways of thinking about ML:

- **Data-first** view: ML is a set of tools for extracting knowledge from data.
- **Deployment-first** (our) view: ML is a set of tools together with a methodology for solving problems using data.

In ML, data is organised as a **dataset** (a collection of **items** described by a set of **attributes**) and knowledge is represented as a **model**.

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

Machine learning taxonomy



Agenda

Problem formulation

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

Examples of regression problems

The following are examples of problems that can be formulated as a regression problem:

1. Predict the energy consumption of a household, given the location of the house, household size, income, intensity of occupation.
2. Predict future values of a company stock, given past stock prices.
3. Predict distance driven by a vehicle given its speed and journey duration.
4. Predict demand given past demand and currency exchange rate.
5. Predict tomorrow's temperature given today's temperature and pressure.
6. Predict the probability to develop a specific heart condition given BMI, alcohol consumption, diet, number of daily steps.

Identify labels and predictors. Do we need machine learning to solve them?

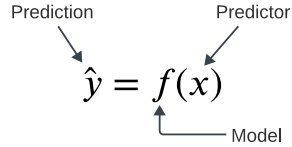
Go to www.menti.com and use code:
DMT: 6505 7423
ICS: 3748 3601

A Salary prediction problem

Let us consider a toy problem; given the age of an individual living in Paris, predict their salary. Here we have collected dataset from 5 individual.

Person (ID)	Age [years]	Salary [$]
S_1	37	68,000
S_2	18	12,000
S_3	66	80,000
S_4	25	45,000
S_5	26	30,000

Mathematical notation



Population:

- x is the **predictor** attribute
- y is the **label** attribute

Dataset:

- N is the number of samples, i identifies each sample
- x_i is the predictor of sample i
- y_i is the actual label of sample i
- (x_i, y_i) is sample i, $\{(x_i, y_i) : 1 \leq i \leq N\}$ is the entire dataset

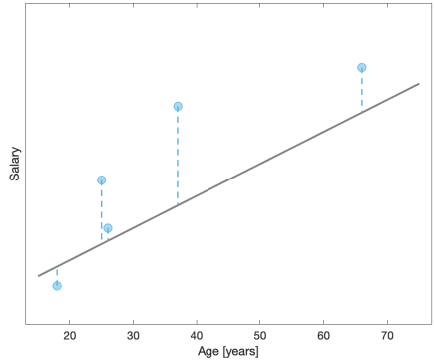
Model:

- $f(\cdot)$ 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

Visualising our mathematical notation

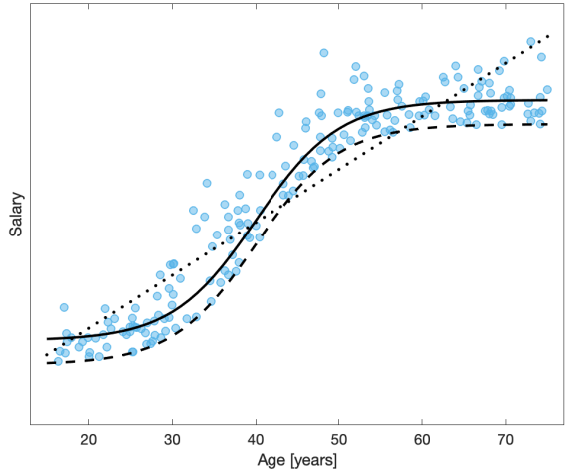
Consider a model $\hat{y} = f(x) = 1000x$

Person (ID)	Age [years]	Salary [£]
S_1	37	68,000
S_2	18	12,000
S_3	66	80,000
S_4	25	45,000
S_5	26	30,000



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**.

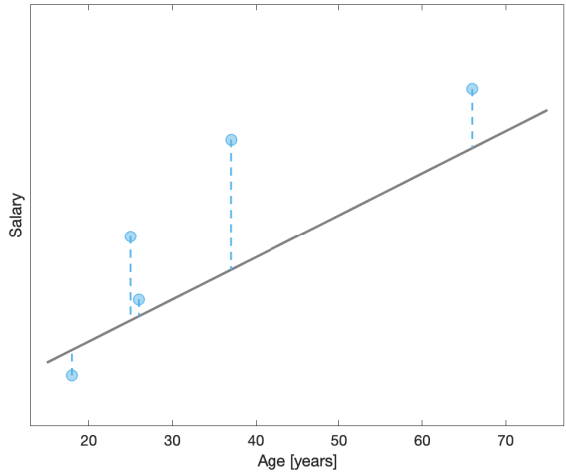
The **squared error** $e_i^2 = (y_i - \hat{y}_i)^2$ is a common quantity used in regression to encapsulate the notion of **single prediction quality**.

Based on the squared error, we can define dataset quality metrics. Two quality metrics based on the squared error are the **sum of squared errors** (SSE) and the **mean squared error** (MSE). which are computed as:

$$E_{SSE} = e_1^2 + e_2^2 + \cdots + e_N^2 = \sum_{i=1}^{N} e_i^2$$

$$E_{MSE} = \frac{1}{N} \sum_{i=1}^{N} e_i^2$$

MSE: Example



Compute E_{SSE} and E_{MSE}

Consider a model $\hat{y} = f(x) = 1000x$, what are the E_{SSE} and E_{MSE} of this model on the given dataset?

Person (ID)	Age [years]	Salary [£]
S_1	37	68,000
S_2	18	12,000
S_3	66	80,000
S_4	25	45,000
S_5	26	30,000

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** 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!**

Choosing a best model

Using E_{SSE} as your quality metric and the given dataset, find the best model among three f_1, f_2, and f_3 as given:

- $f_1(x) = 1000x$
- $f_2(x) = 999x$
- $f_3(x) = 1000 + 1000x$

Person (ID)	Age [years]	Salary [£]
S_1	37	68,000
S_2	18	12,000
S_3	66	80,000
S_4	25	45,000
S_5	26	30,000

Regression as an optimisation problem (*take 1*)

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}:

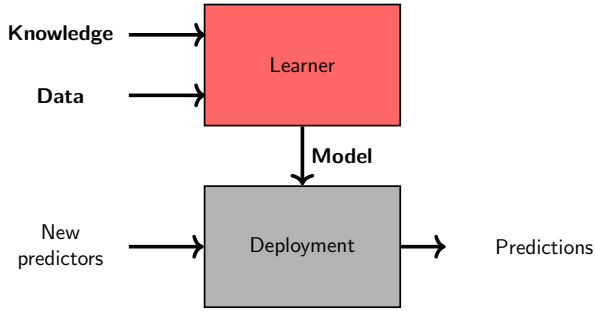
$$f_{best}(x) = \arg\min_f \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2$$

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

Why *take 1*? Note that we are defining regression as finding the model that minimises E_{MSE} *on the dataset*, without considering what happens *once deployed*. We'll revise this definition.

Agenda

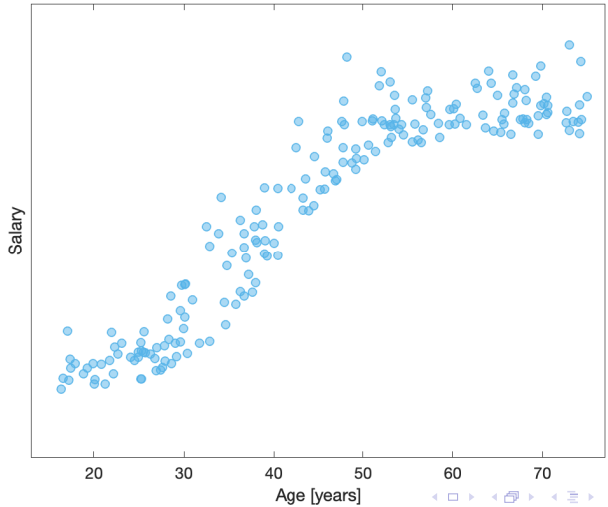
Our regression learner



- **Dataset**: Labelled samples (predictors and true 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

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

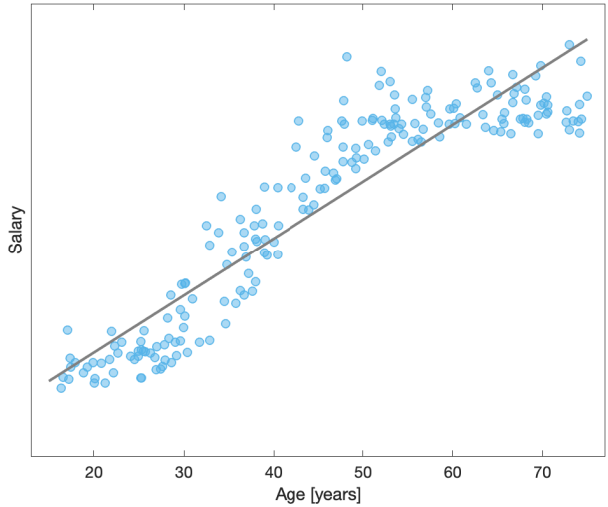
Hence, the predicted label $\hat{y}_i$ can be expressed as

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

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

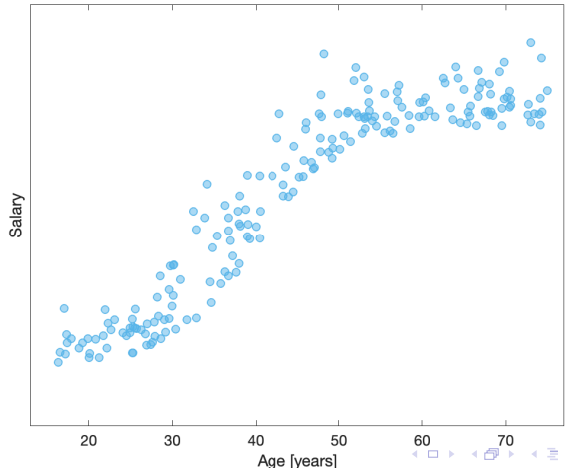
In machine learning, we use a **dataset** to tune the parameters. We say that we **train the model** or **fit the model** to the **training dataset**.

Linear solution: Example



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:

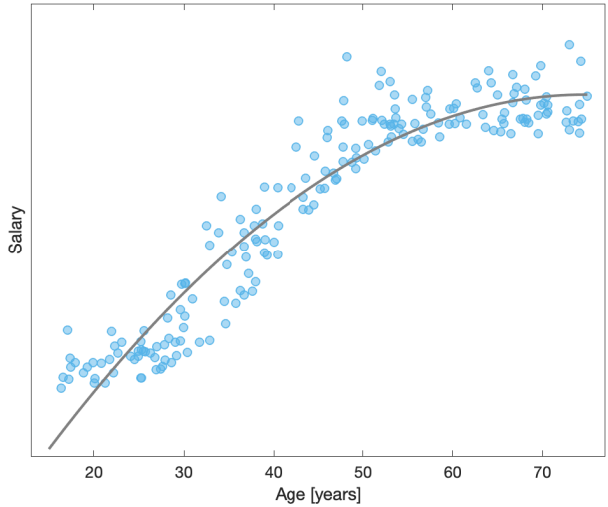
$$f(x_i) = w_0 + w_1 x_i + w_2 x_i^2 + \cdots + w_D x_i^D$$

where D is the degree of the polynomial.

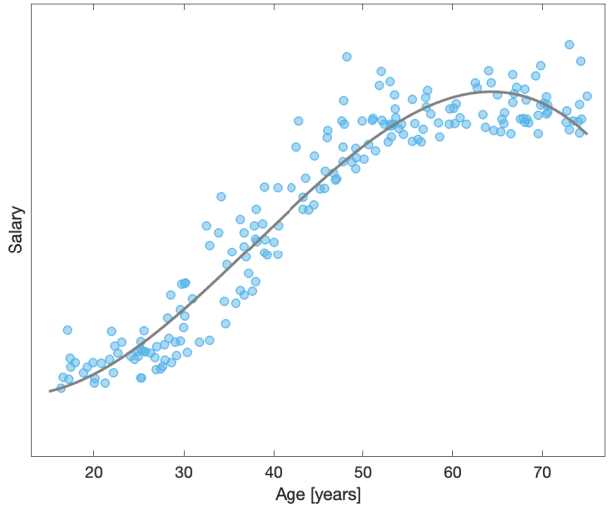
Polynomial regression defines a **family of families** of models. For each value of D, we have a different family: $D = 1$ corresponds to the linear family, $D = 2$ to the quadratic, $D = 3$ to the cubic, and so on.

We call D a **hyperparamenter**. What it means is that setting its value results in a different family, with a different collection of parameters.

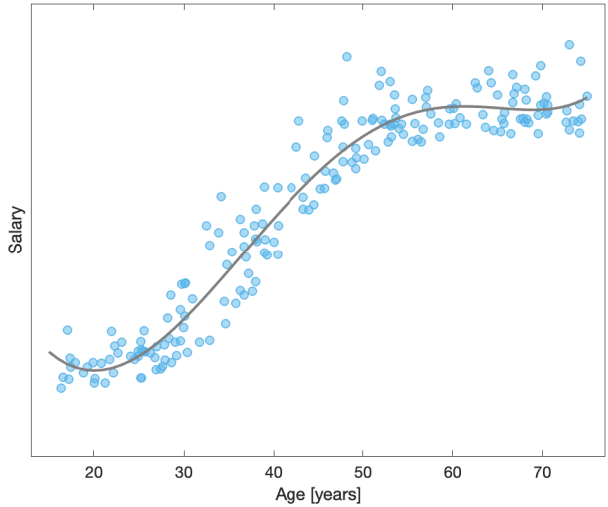
Quadratic solution



Cubic solution

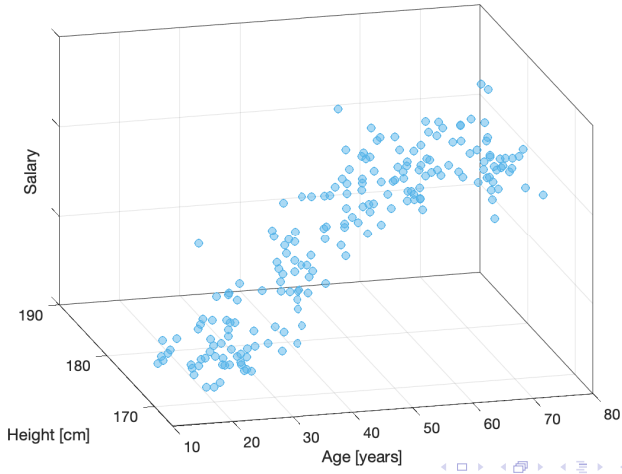


5-power solution



Multiple regression

In multiple regression there are **two or more predictors**. Given item i, we will denote each individual predictor as $x_{i,1}$, $x_{i,2}$, ... and $x_{i,K}$, where K is the number of predictors.



Multiple regression: Linear model

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

Multiple regression: Vector notation

Let K denote the number of predictors. We will represent the k-th predictor of item i as $x_{i,k}$.

Using vector notation, the predictors of item i can be packed together into a vector represented in **bold font**:

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

where the constant 1 is prepended for convenience.

Using vector notation, 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

Linear models in multiple regression are simply the sum of a constant (or intercept) and each predictor multiplied by its own coefficient.

Multiple linear regression models can be expressed as:

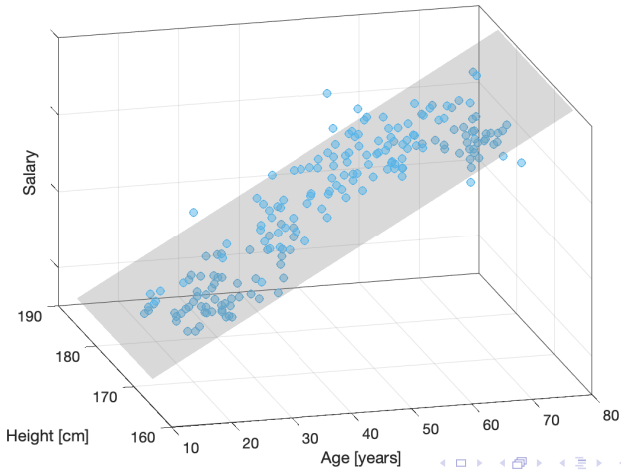
$$f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i = w_0 + w_1 x_{i,1} + \cdots + w_K x_{i,K}$$

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** $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,K} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,K} \end{bmatrix}$$

together with the **label vector** $\boldsymbol{y}$:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Multiple regression: More notation

Given a linear model defined by coefficients

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{bmatrix}$$

we can calculate the **label vector** $\hat{\boldsymbol{y}}$ as

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \mathbf{X}\boldsymbol{w} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,K} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,K} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{bmatrix}$$

and error vector $\boldsymbol{e}$ as

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$$

Multiple linear regression: Example

Consider a dataset consisting of 4 samples described by three attributes:

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

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

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:

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

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

This solution can also be used for **polynomial models**, by treating the powers of the predictor as predictors themselves.

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

The least squares solution: ~~Example 1~~

$X, y \quad \hat{y} = Xw$

$f(x) = x^2 y + x + 1$

$\frac{\partial f}{\partial x} = 2xy + 1$

$w = ? \Longrightarrow E_{MSE}$ - minimum

$\frac{\partial f}{\partial x} = 0$

$2xy + 1 = 0$

$x = -\frac{1}{2y}$

The least squares solution: Example-2

$X, y \qquad \hat{y} = Xw$

$$E_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$E = \frac{1}{N}(y - \hat{y})^T(y - \hat{y})$$

$$E = \frac{1}{N}(y - Xw)^T(y - Xw)$$

$e_1\ e_2\ e_3\ \ldots\ e_{16}$

$\sum e_i^2$

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \\ e_{16} \end{bmatrix}$$

$e^T e$

The least squares solution: Derivation

$$E = \frac{1}{N}(y^T - w^T X^T)(y - Xw)$$

$$= \frac{1}{N}(y^T y - y^T Xw - w^T X^T y + w^T X^T Xw)$$

$$E = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T Xw)$$

$$\frac{\partial E}{\partial w} = \frac{1}{N}(0 - 2X^T y + 2X^T Xw)$$

Other models for regression

$$\frac{\partial E}{\partial w} = \frac{-2}{N}\left(X^T y - X^T X w\right)$$

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

$$\frac{\partial E}{\partial w} = 0 = \frac{-2}{N}\left(X^T y - X^T X w\right)$$

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

$$X^T y = X^T X w$$

$$w = (X^T X)^{-1} X^T y$$

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

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.

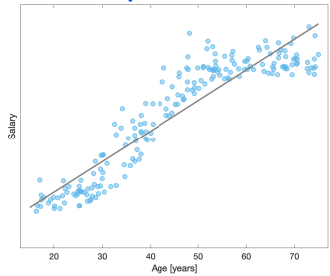
The flexibility of a model is related to its **interpretability** and **accuracy** and there is a trade-off between the two.

Interpretability

Explainability

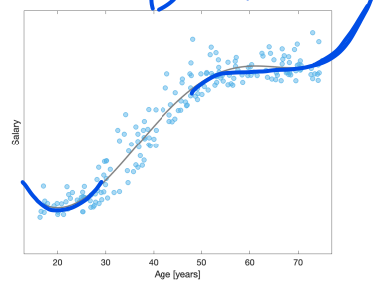
Model interpretability is crucial for us, as humans, to understand in a qualitative manner how a predictor is mapped to a label. Inflexible models produce solutions that are usually simpler and easier to interpret.

D=1



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

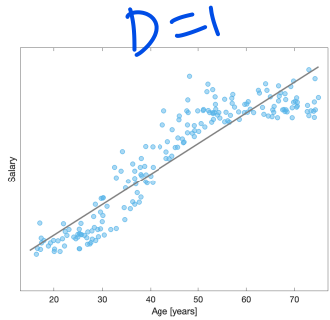
D=5

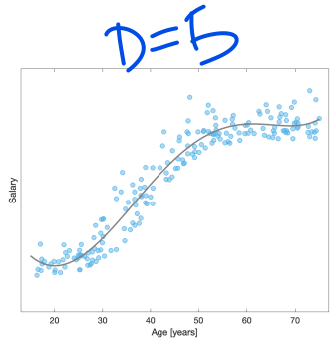


According to this polynomial model, our salary remains the same as teenagers, then increases between our 20s and 50s, then...

Quality on the training dataset

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



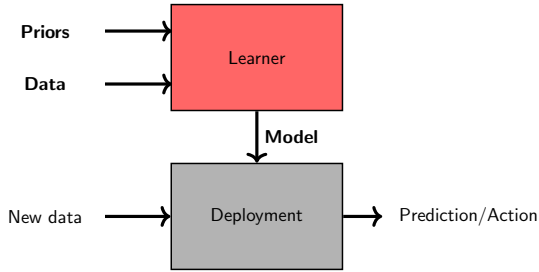


The **training error** of the best linear model is $E_{MSE} = 0.0983$

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

Generalisation

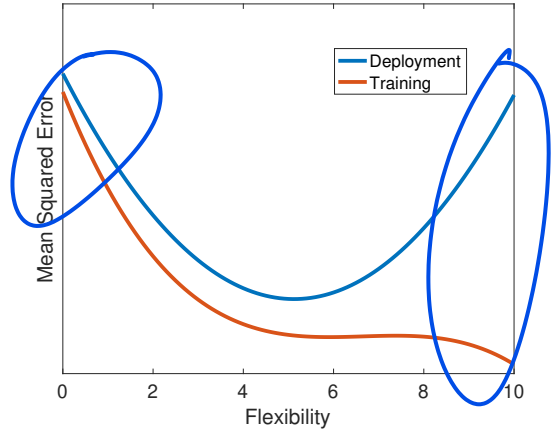
We have considered the **training MSE**, i.e. the quality of regression models on the **training dataset**.



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?

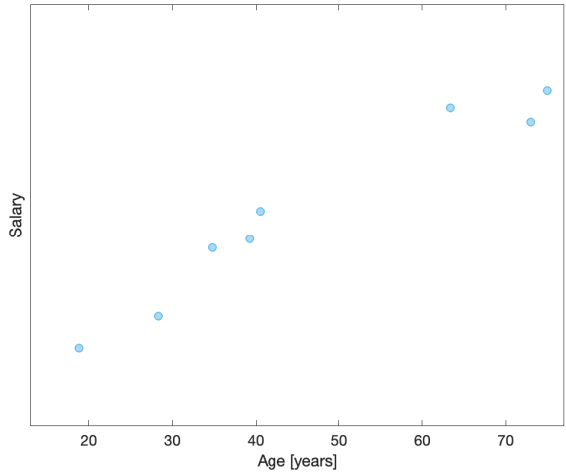


Underfitting and overfitting

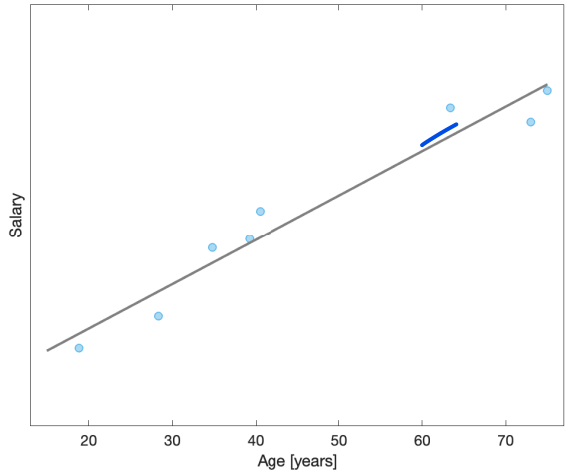
By comparing the performance of models during training and deployment, we can observe three different behaviours:

- **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**.

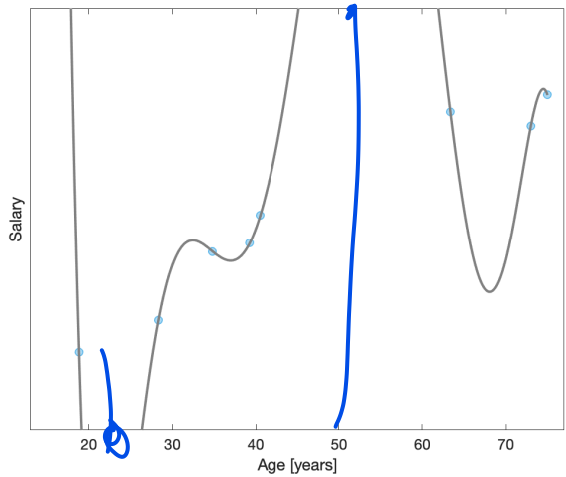
Underfitting and overfitting



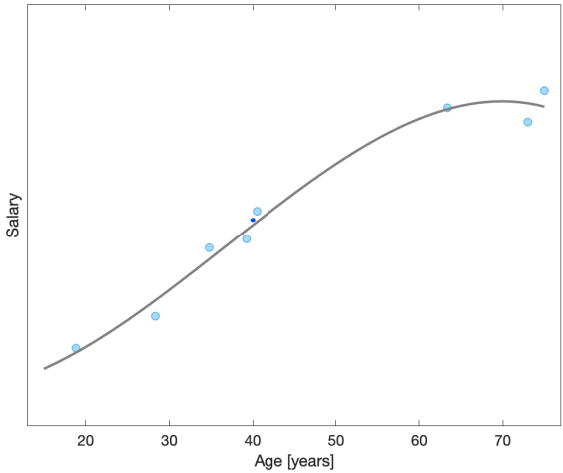
Underfitting



Overfitting

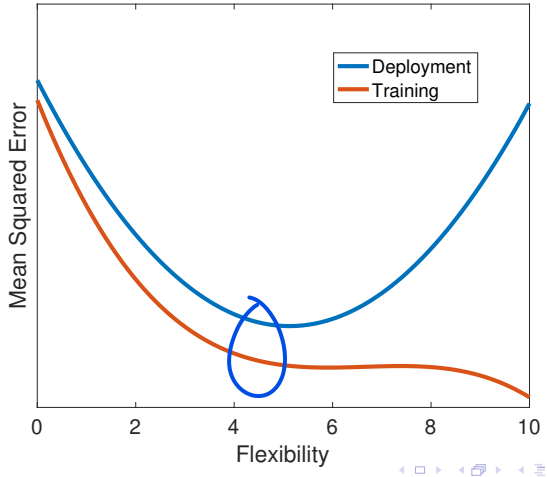


Just right

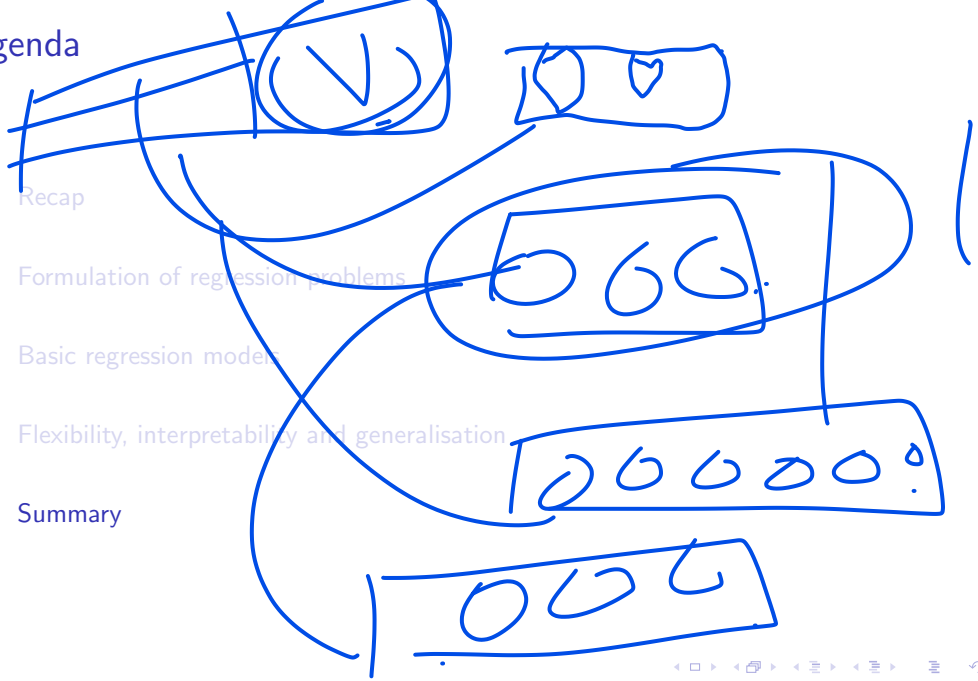


Underfitting and overfitting

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*.

Queen Mary
University of London