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

ECS766 Data Mining
Week 2: Regression

Dr Jesús Requena Carrión

2 Oct 2019

Agenda

Recap (with some extras)

Formulation of regression problems

Basic regression models

Flexibility, overfitting and regularisation

Final remarks

Data Mining

Some data scientists define Data Mining as the art of extracting **knowledge** (i.e. building a model) from **existing data** (anything that has been recorded).

The term **mining** suggests that

- Knowledge has **value**, as models can be sold and generate revenue when deployed.
- Data **already exist** and are waiting for us to find its hidden value.

So someone has already collected our data, great! Great?

The 1936 Literary Digest poll

Alfred Landon
Republican Party

Franklin D. Roosevelt
Democratic Party

- The Literary Digest conducted one of the largest and most expensive polls ever conducted (around 2.4 million people)
- Predicted Landon would get 57 % of the vote, Roosevelt 43 %
- The actual results of the election were 62 % for Roosevelt against 38 % for Landon (19 % error, the largest ever)
- The cause: **Bad sampling**, 10 million names were taken from telephone directories, club membership lists, magazine subscribers lists, etc. The poll suffered from **selection** and **nonresponse bias** and samples were **not representative** of the population.

Know thy data!

The dataset as a table

Datasets can be represented as tables, where **rows correspond to instances** (a.k.a. *samples*) and **columns to attributes** (a.k.a. *features*).

The first 5 instances of a dataset recording the age and salary of a group of people are shown below in a table form:

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

Matlab tabular data structures are simply called **Tables** and in Python's Pandas library they are known as **Dataframes**.

The dataset as a matrix

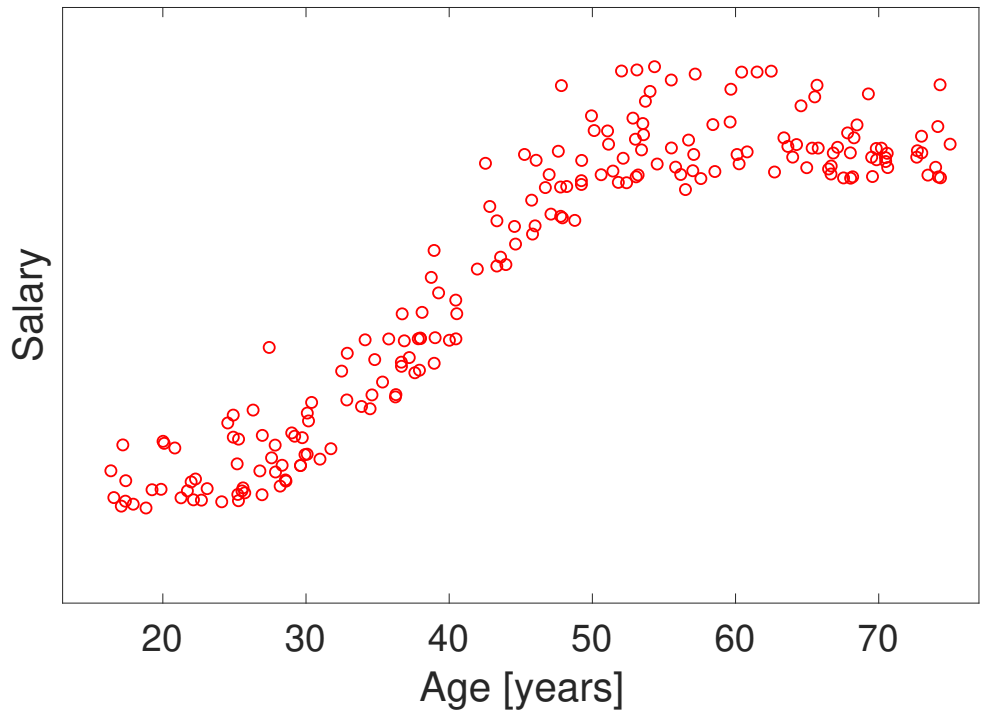
Numerical attributes can also be represented in matrix form, for instance:

$$S = \begin{pmatrix} 18 & 12000 \\ 37 & 68000 \\ 66 & 80000 \\ 25 & 45000 \\ 26 & 30000 \\ \vdots & \vdots \end{pmatrix}$$

This is a useful and compact notation that will make it easier for us to formulate problems and represent computations. In Matlab and Python's Numpy library, we can use **array** data structures to represent matrices.

The dataset as a point cloud

Datasets can also be represented as a set of points in a space. For every instance, the value of one feature is represented against the value of the other features in a cartesian coordinate system.



Agenda

Recap (with some extras)

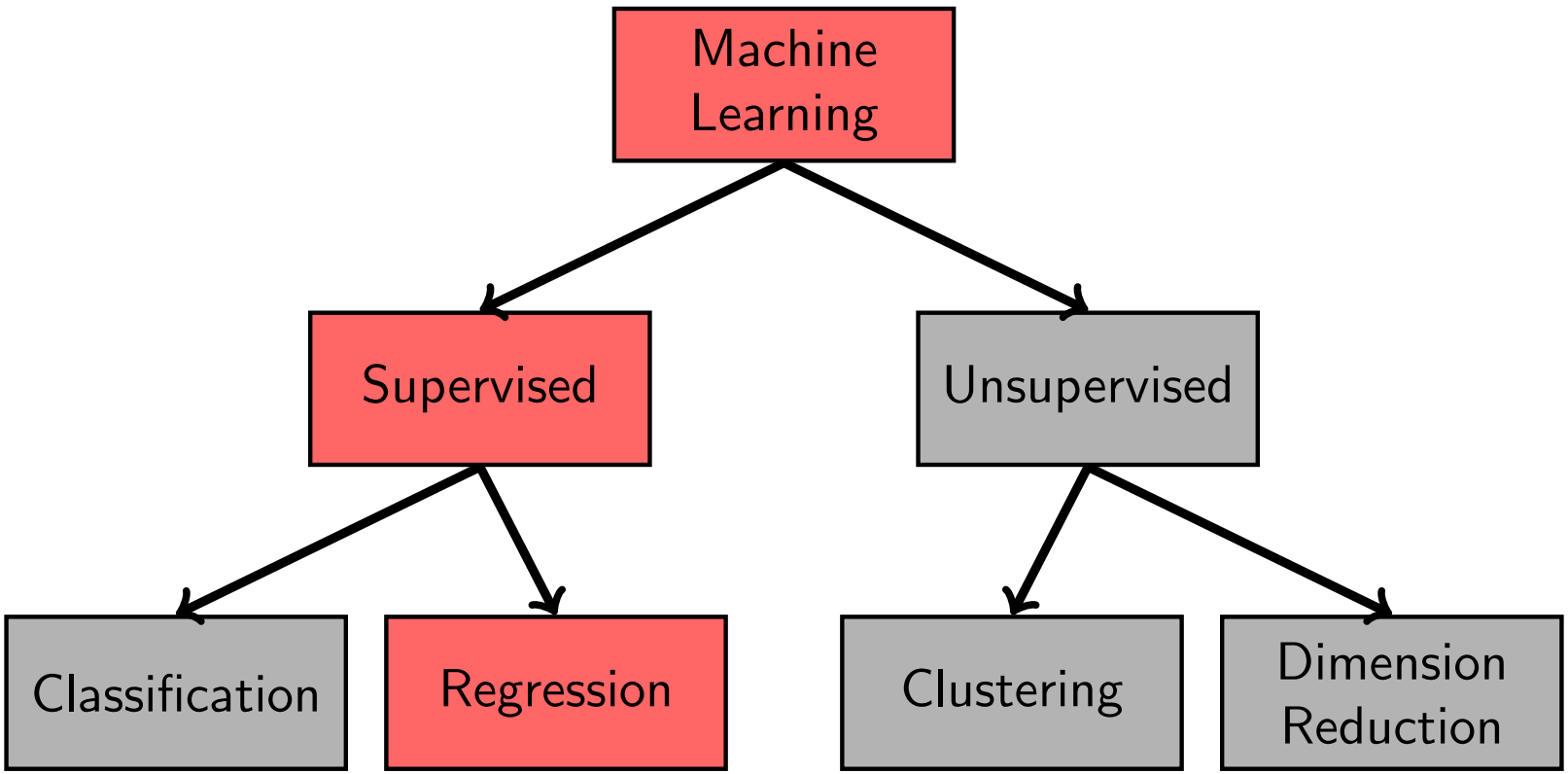
Formulation of regression problems

Basic regression models

Flexibility, overfitting and regularisation

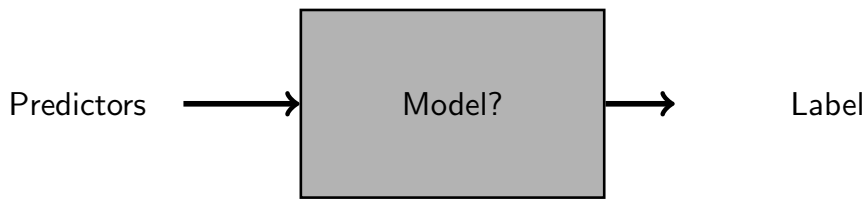
Final remarks

Data science taxonomy



Problem formulation

- **Supervised**: Our starting point is the assumption that the value of one of the attributes of our dataset (the label) can be predicted from the value of the remaining attributes (the 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.



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:

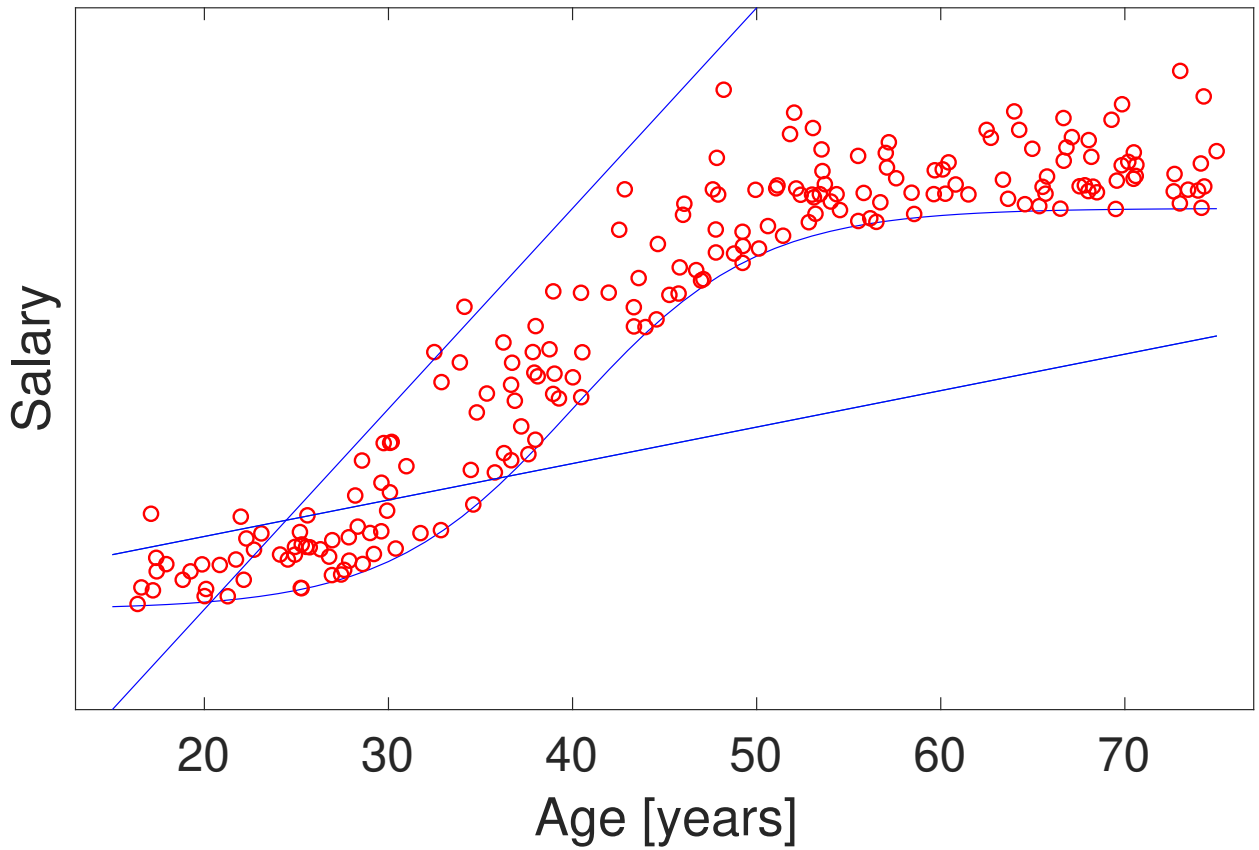
(a) *Age* is the predictor, *Salary* is the label

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

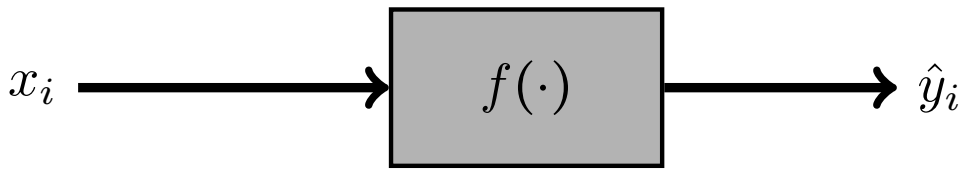
(c) Both options can be considered

Examples of candidate solutions for our dataset

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



Mathematical notation



Dataset:

- N is the number of samples in our dataset
- i identifies one of the samples
- x_i is the predictor of sample i
- y_i is the label of sample i
- The dataset is $\{(x_i, y_i) : 1 \leq i \leq N\}$

Model:

- $f(\cdot)$ denotes our model
- $\hat{y}_i = f(x_i)$ is the label produced by $f(\cdot)$ when the predictor is x_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.*)

What is a good model?

In order for us to find the **best** model we need to have a quantitative notion of **model quality**, which can be quantified by a **goodness of fit**, **loss function** or **error function**.

One popular option is the **mean square error** (MSE), which is defined 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 complex enough model achieving this

The nature of the error

In general, when considering a regression problem we need to be aware that:

- The chosen **predictors might not reflect all the factors** that determine a label
- A chosen **model might not be able to represent accurately** the true relationship between response and predictor
- **Random mechanisms** might be involved too

We represent this discrepancy mathematically as

$$y = f(x) + e$$

In words, we accept that there will always be some discrepancy (error e) between the desired response y and our model f. **Embrace the error!**

Regression as an optimisation problem

Given a dataset $\{(x_i, y_i) : 1 \leq i \leq N\}$, each possible model f has a corresponding E_{MSE}. Our problem is then to **find the model f with the lowest MSE**:

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

The question is, **how** do we find such model? This is an **optimisation problem** and the solution for this problem is called the **minimum mean square error** (MMSE) solution.

Agenda

Recap (with some extras)

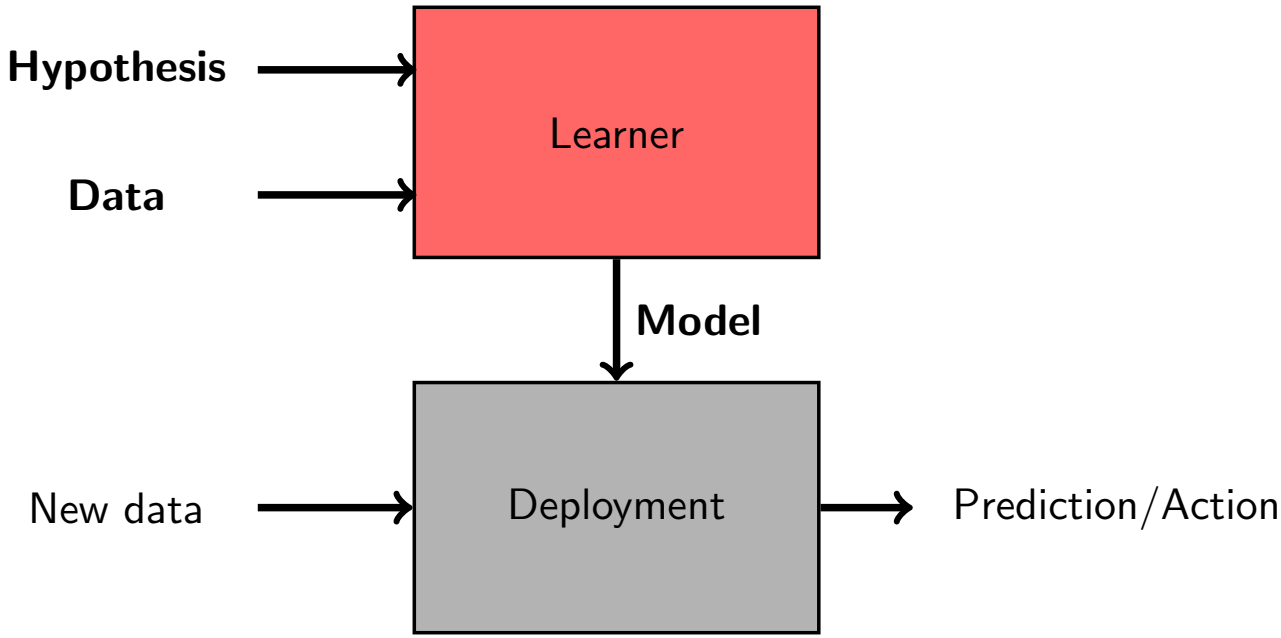
Formulation of regression problems

Basic regression models

Flexibility, overfitting and regularisation

Final remarks

Our regression learner



- **Hypothesis**: Type of model (linear, polynomial, etc). **Data exploration** can help us choose the type of model
- **Data**: In the simplest case, each sample has two attributes (one **predictor** and one **continuous label**)
- **Model**: Produces one label in response to a given predictor. It is determined by finding (**optimisation**) the best model (**evaluation**) within a family of models.

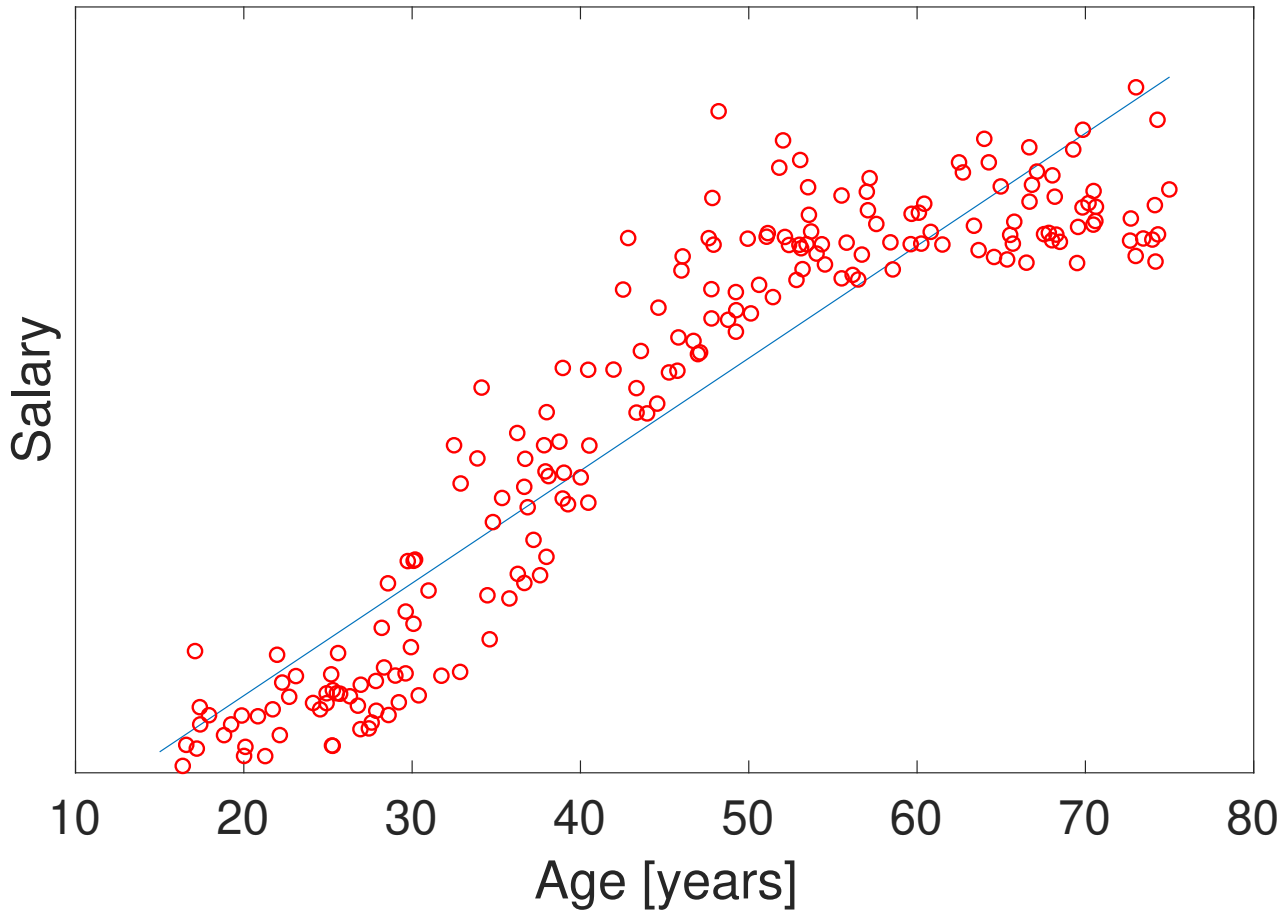
Simple linear regression

In simple linear regression, one predictor x and one label y are considered and models are defined by the mathematical expression

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

Linear models are therefore *parametric* and have two parameters w_0 and w_1, which need to be *tuned*. The *best* values for w_0 and w_1 are the ones that minimise the loss function (MSE, for instance).

MMSE linear solution: Example



Multiple linear regression: Notation

In multiple regression, there are two or more predictors. For instance, we could use the age and height of individuals to predict their salary.

Using vector notation, predictors can be expressed as a vector

$$\boldsymbol{x} = [1, x_1, x_2, \ldots, x_P]^T,$$

where x_p denotes the p-th predictor, P is the number of predictors and the constant 1 is appended for convenience.

A multiple regression model can then be expressed as the function

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

Note that all the previous definitions and derivations (error, MSE, etc) can be readily translated to the multivariate scenario.

Multiple linear regression: Notation

We will use the symbol $\boldsymbol{x}_i$ to denote the i-th sample in a dataset,

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

where $x_{i,p}$ is the p-th predictor of the i-th sample. The whole dataset can be represented by the following matrix X and vector $\boldsymbol{y}$:

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

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

Multiple linear regression: Formulation

Using the proposed vector notation, linear multiple models can be expressed as:

$$f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{w} = w_0 + w_1 x_1 + \cdots + w_P x_P$$

where $\boldsymbol{w} = [w_0, w_1, \ldots, w_P]^T$ 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 $\boldsymbol{x} = [1, x]^T$.

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

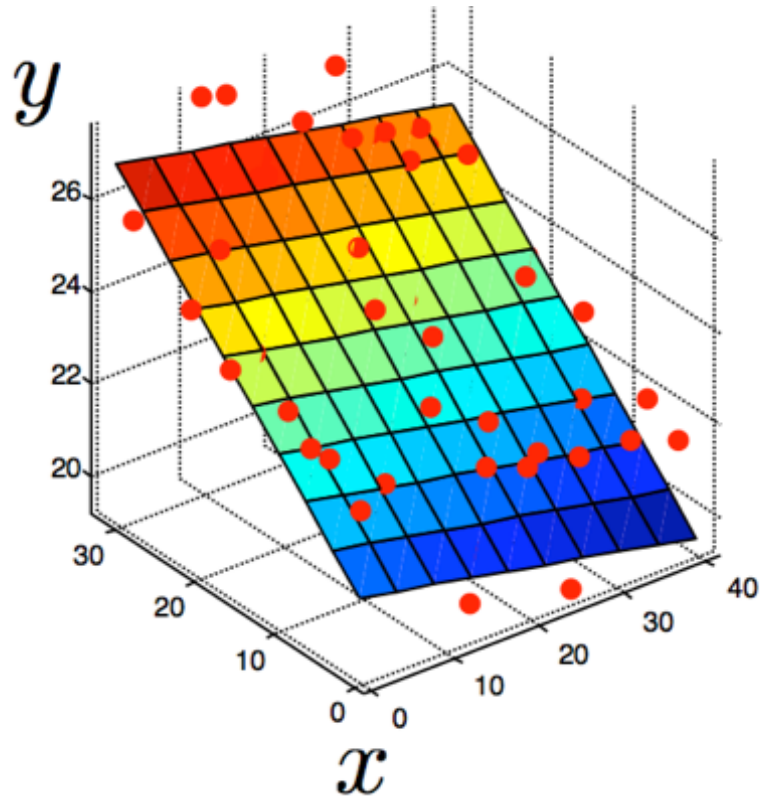
Consider a dataset consisting of 4 samples described by three attributes, namely age, height and salary:

	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 each predictor in the dataset and the linear regression model.
2. Define a matrix X and a vector $\boldsymbol{y}$ containing respectively the predictors and labels in the dataset.

Multiple linear regression



The MMSE solution for linear regression models

It can be shown that the linear model that minimises the MSE (i.e. the MMSE solution) is defined by the coefficients

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

This is an **exact** or **analytical solution**. Note that the inverse matrix $\left(X^T X\right)^{-1}$ exists when all the columns in X are independent.

In general there will not exist an analytical expression that allows us to calculate the optimal parameters of a model. Instead, we will need to use **numerical optimisation** (gradient descent, evolutionary algorithms, grid search...) to find the optimal parameters of a model.

Simple polynomial regression

The general form of a polynomial regression model is:

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

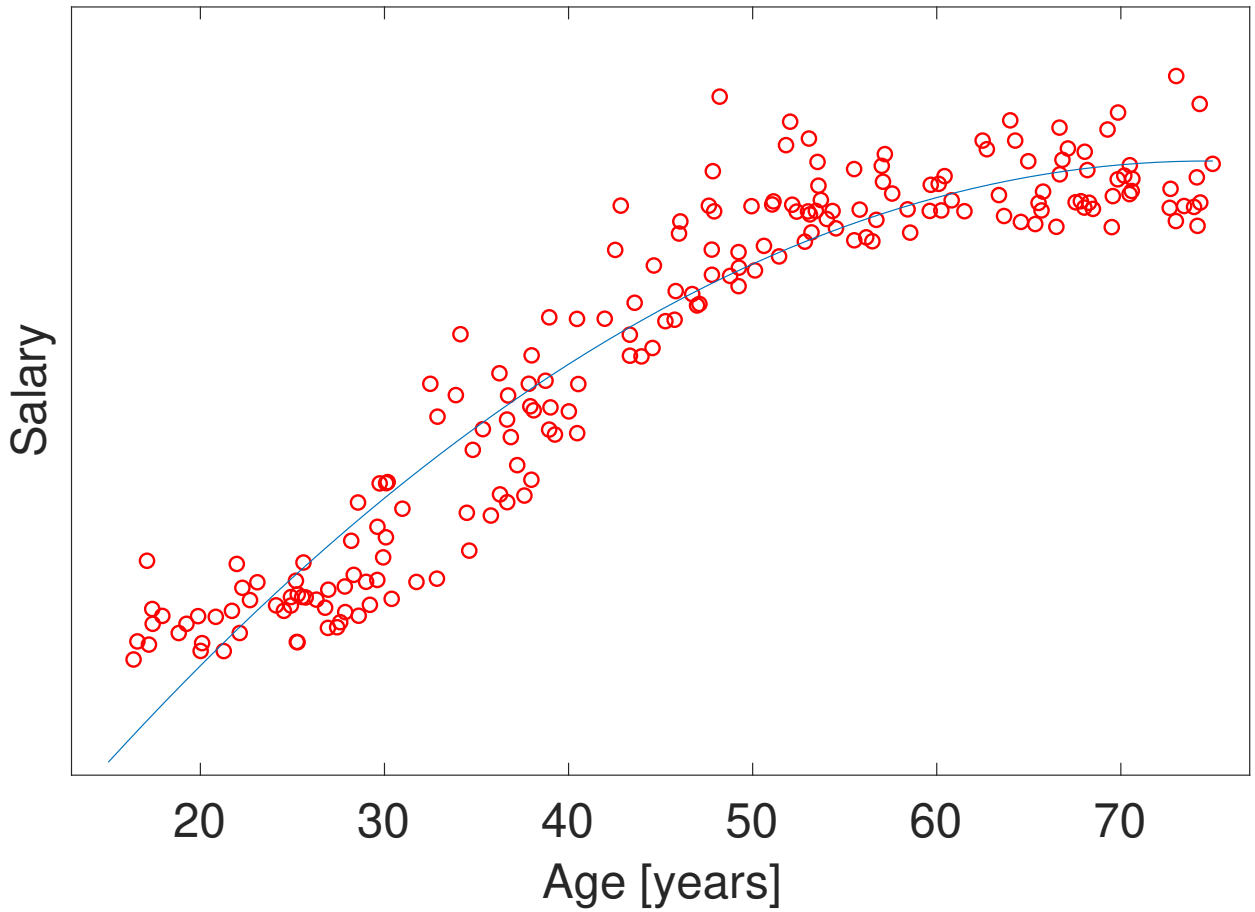
where D is the degree of the polynomial.

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

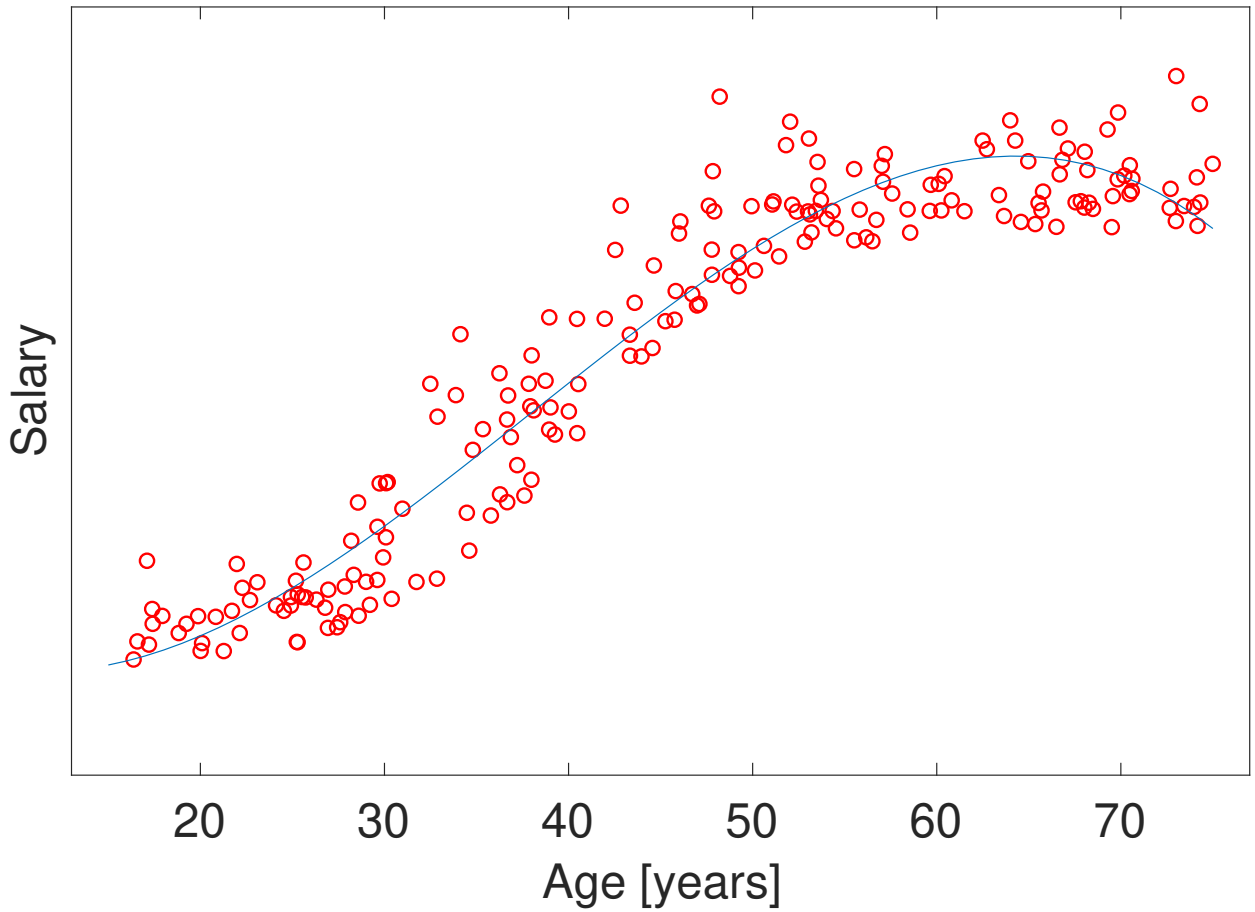
$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 = \boldsymbol{w}^T \boldsymbol{\phi}$$

where $\boldsymbol{\phi} = [1, x, x^2, x^3]^T$. Therefore, there exists an exact MMSE solution for simple polynomial regression (see previous slide).

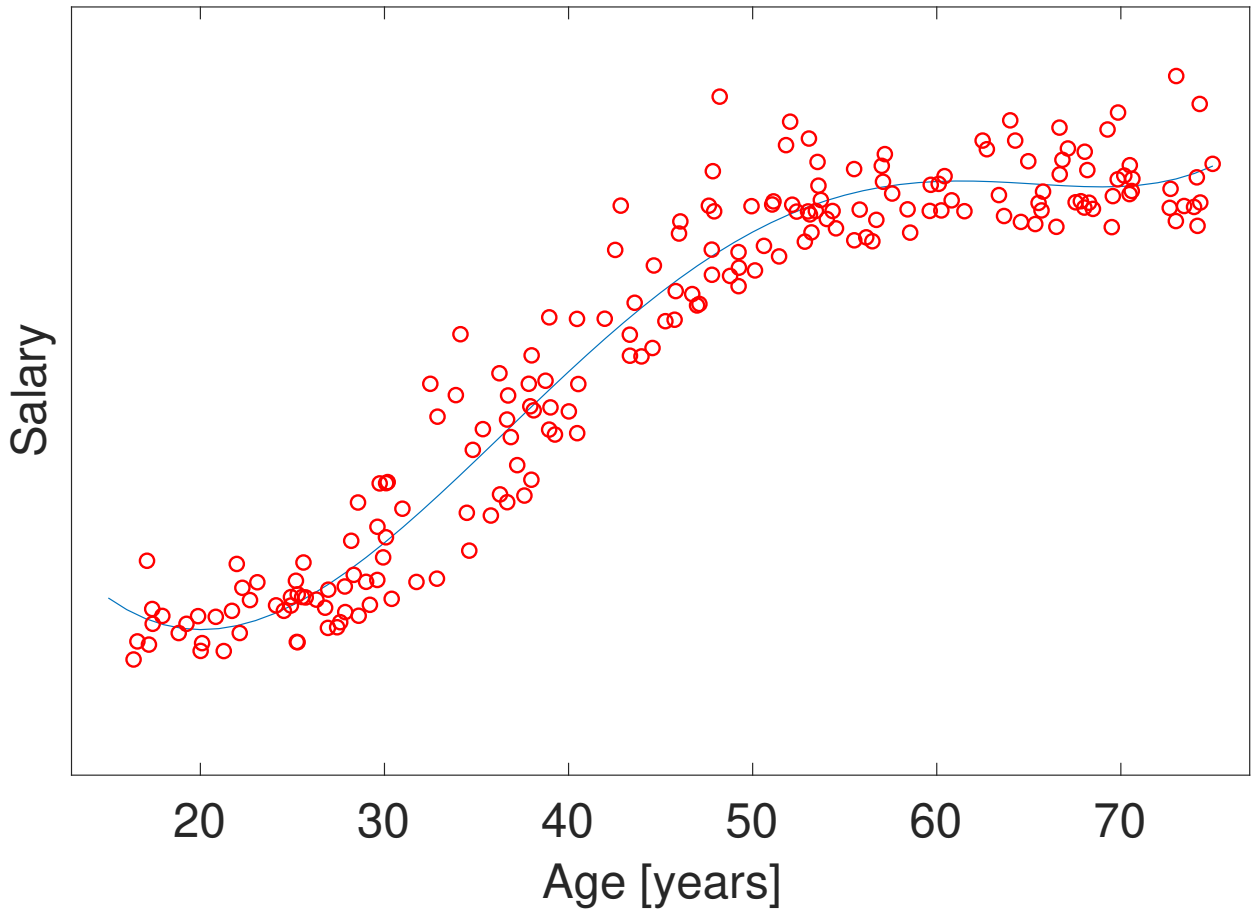
MMSE quadratic solution



MMSE cubic solution



MMSE 5-power solution



Agenda

Recap (with some extras)

Formulation of regression problems

Basic regression models

Flexibility, overfitting and regularisation

Final remarks

Flexibility

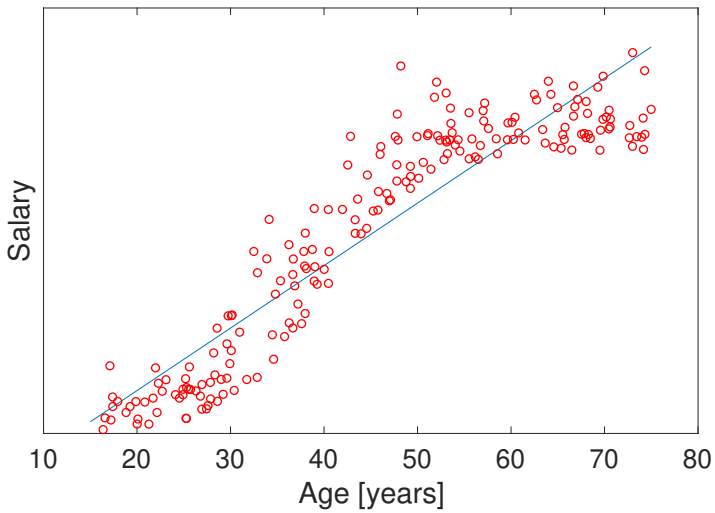
Flexible models allow us to generate multiple shapes by tuning their parameters. Sometimes, we talk about the **degrees of freedom** or **complexity** to describe their flexibility. The degrees of freedom of a model are in general related to their number of parameters.

- Linear models are inflexible, as they can only generate straight lines. They have only 2 parameters.
- Cubic models are more flexible and are characterised by 4 parameters.

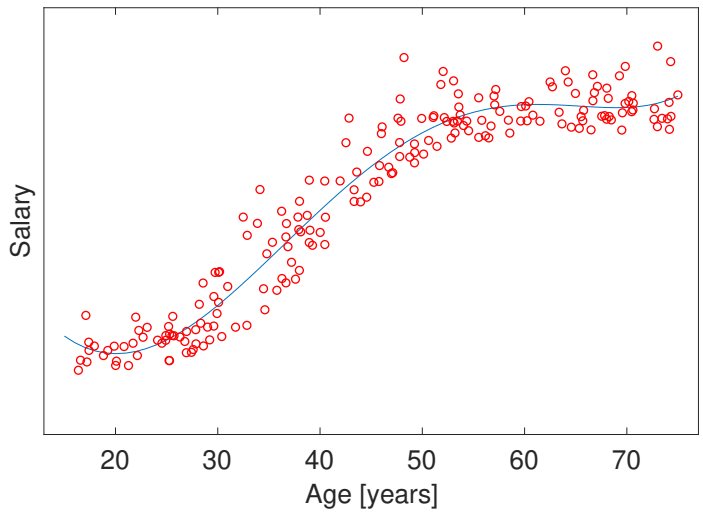
The flexibility of a model is related to their **interpretability** and **accuracy** and there is a trade-off between the two.

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.



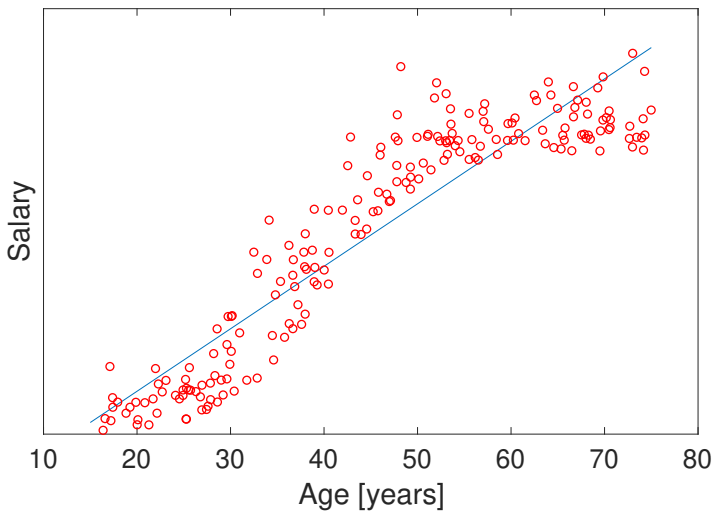
According to this linear model, the older you get, the more money you make



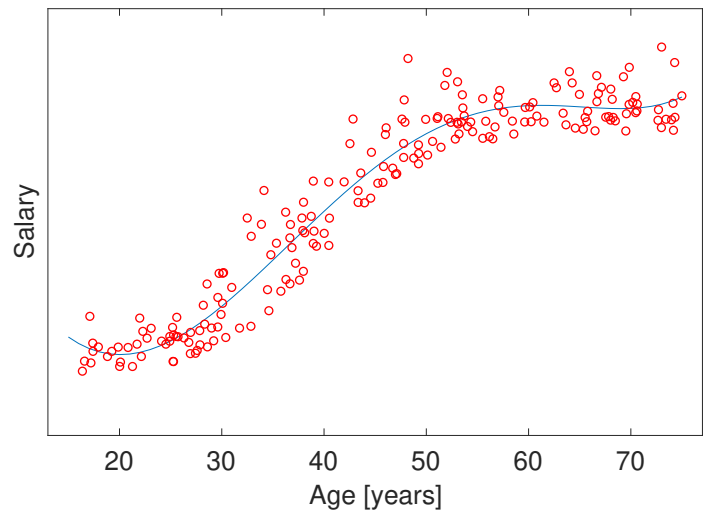
According to this polynomial model, our salary remains the same as teenagers, then increases between our 20s and 50s, then...

Accuracy

The accuracy of a model is also related to its flexibility. Flexible models will have in general lower MSE than inflexible models.



The error for the MMSE linear model is $E_{MSE} = 0.0983$



The error for this MMSE polynomial model is $E_{MSE} = 0.0379$

Generalisation

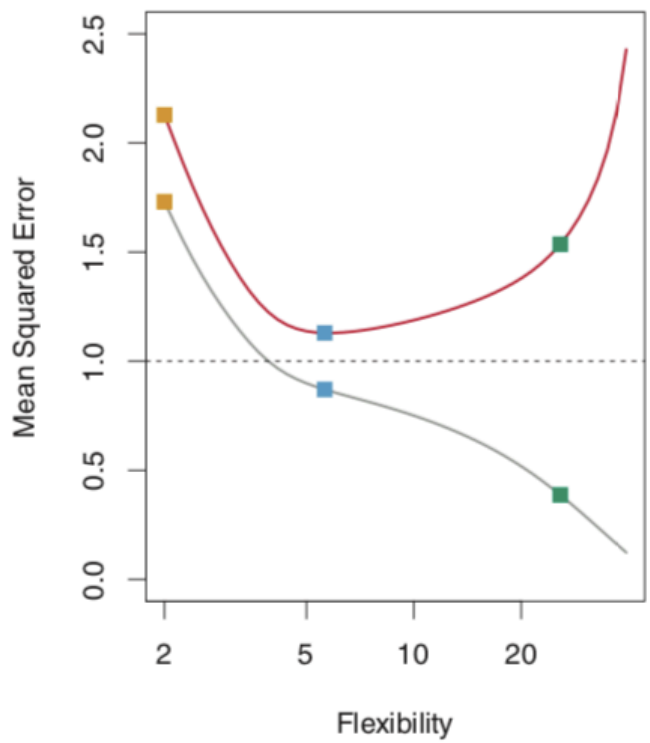
As data miners, we use datasets to build models that will be deployed. Hence, our aim is not to create a model that works well for our dataset, but during production.

So far, we have assessed the quality of our model by computing the E_{MSE} on the dataset used to build the model. How can we be sure our model works well during production?

Generalisation is the ability of our model to work well in production, in other words, to successfully **translate what we have learnt during the learning stage to the production stage**.

Generalisation

In this figure, the grey curve represents the MSE obtained during learning for models of increasing complexity, whereas the red curve represents the MSE during production for the same models. What's happening?



Taken from *An Introduction to Statistical Learning* by G. James et al.

Training and test

In order to evaluate the quality of our final model, we split our dataset into two disjoint datasets, the **training** dataset and the **test dataset**:

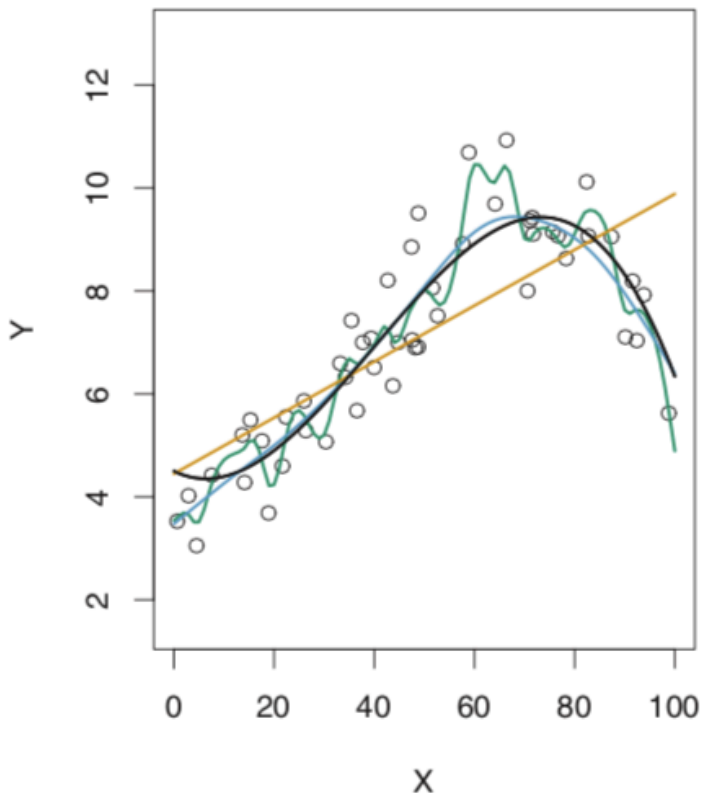
- The training dataset is used to **build** our model
- The test dataset is used to **evaluate** the final model (remember to do this only once or the **Infinite Monkey Theorem** will get you!)

Accordingly, we have two different errors:

- The **training error** is the model's error calculated on the training dataset
- The **test error** is the model's error calculated on the test dataset

Overfitting

Our model is **overfitting** when it produces small training MSE and large test MSE. Overfitting is due to too complex models and not enough data. By contrast, **underfitting** is characterised by large training and test MSE.



Taken from *An Introduction to Statistical Learning* by G. James et al.

Who is overfitting / underfitting in the following figure?

(a) Green is underfitting, black is overfitting

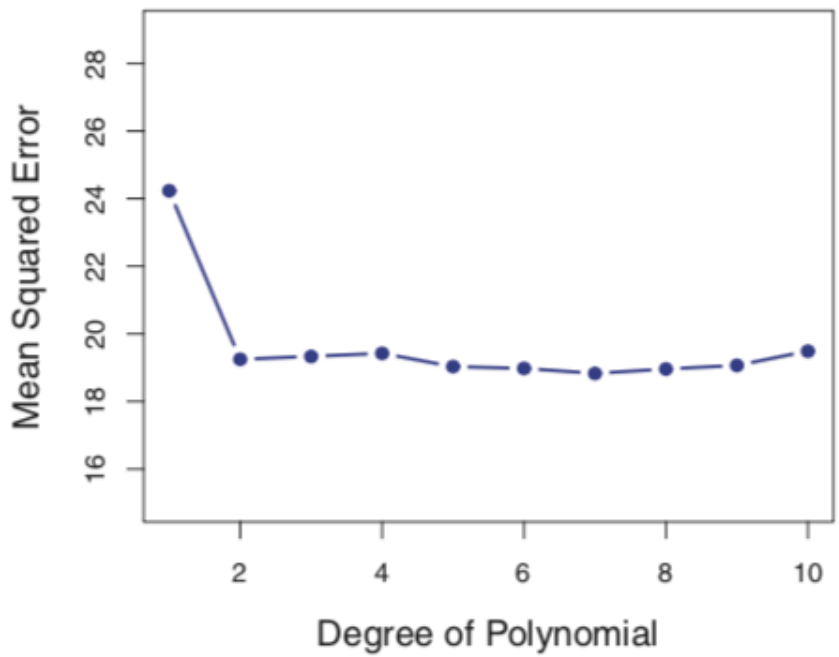
(b) Yellow is underfitting, black is overfitting

(c) Yellow is underfitting, green is overfitting

Validation

As data miners we have a wide range of models to chose from. Take the family of polynomial models, which degree should we choose?

Validation methods allow us to select the complexity of our model. The simplest approach is to split our dataset into training and validation datasets and use the validation error to choose the model complexity.



Regularisation

Regularisation is a procedure for reducing the risk of model overfitting. The idea is to create a goodness function that penalises large coefficients in a linear model $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$:

$$E_{MSER} = \frac{1}{N} \sum_{i=1}^{N} e_i^2 + \lambda w^T w$$

The solution $\boldsymbol{w}$ that minimises E_{MSER} is

$$\boldsymbol{w} = \left(X^T X + N\lambda I \right)^{-1} X^T \boldsymbol{y}$$

When $\lambda = 0$, we obtain the solution for the usual MSE problem. As λ increases, the complexity of the resulting solution decreases and so does the risk of overfitting. However, notice that the parameter λ still needs to be determined (by using some form of validation).

Agenda

Recap (with some extras)

Formulation of regression problems

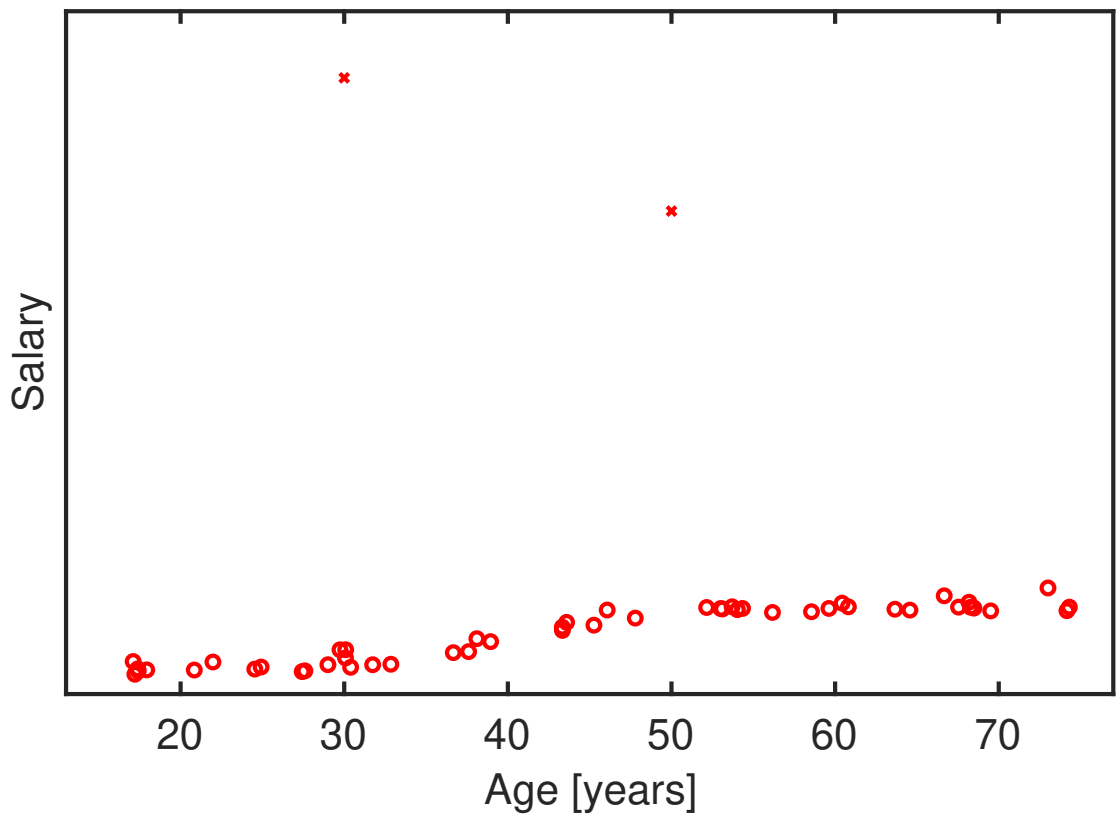
Basic regression models

Flexibility, overfitting and regularisation

Final remarks

Outliers

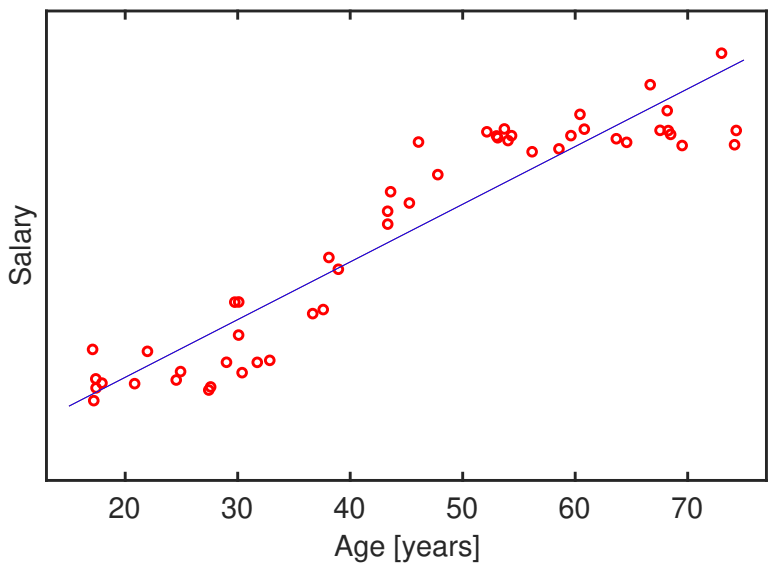
Outliers are samples that do not follow the general pattern of your dataset. In a graph representation, they can be seen abnormally distant for the rest of the samples.



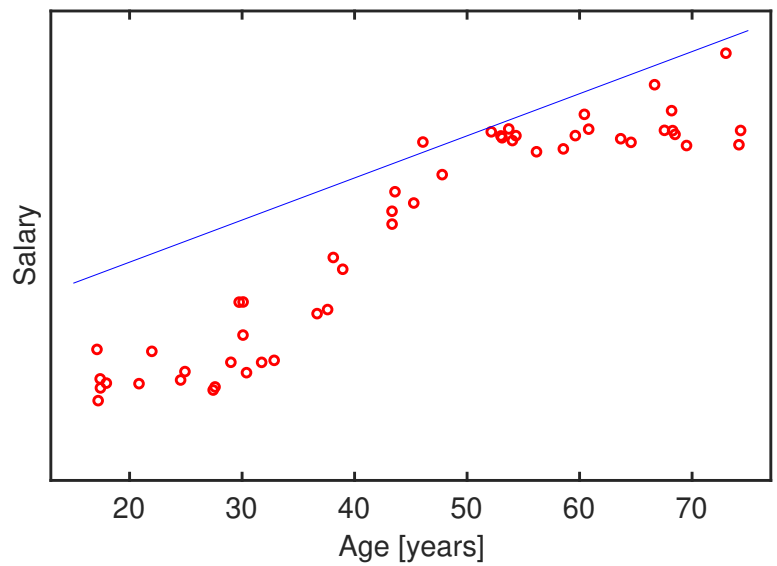
Outliers

Outliers can have a big impact on our solution and therefore, they are identified and eliminated during the stage of data exploration.

Without outliers



With outliers



Other error functions

In addition to the MSE, there are other quality measures:

- **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.

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

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

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

Other models

Other models that can be used include:

- Exponential
- Sinusoids
- Neural networks
- Radial basis functions
- Splines
- ...

Do we always need data-driven approaches?

Think about this: Would you use data-driven approaches to build a model that predicts the distance driven by a car moving at a constant speed?