dm25s1

Topic 06 : Data Modelling

Part 01 : Data Modelling - Introduction

Dr Bernard Butler

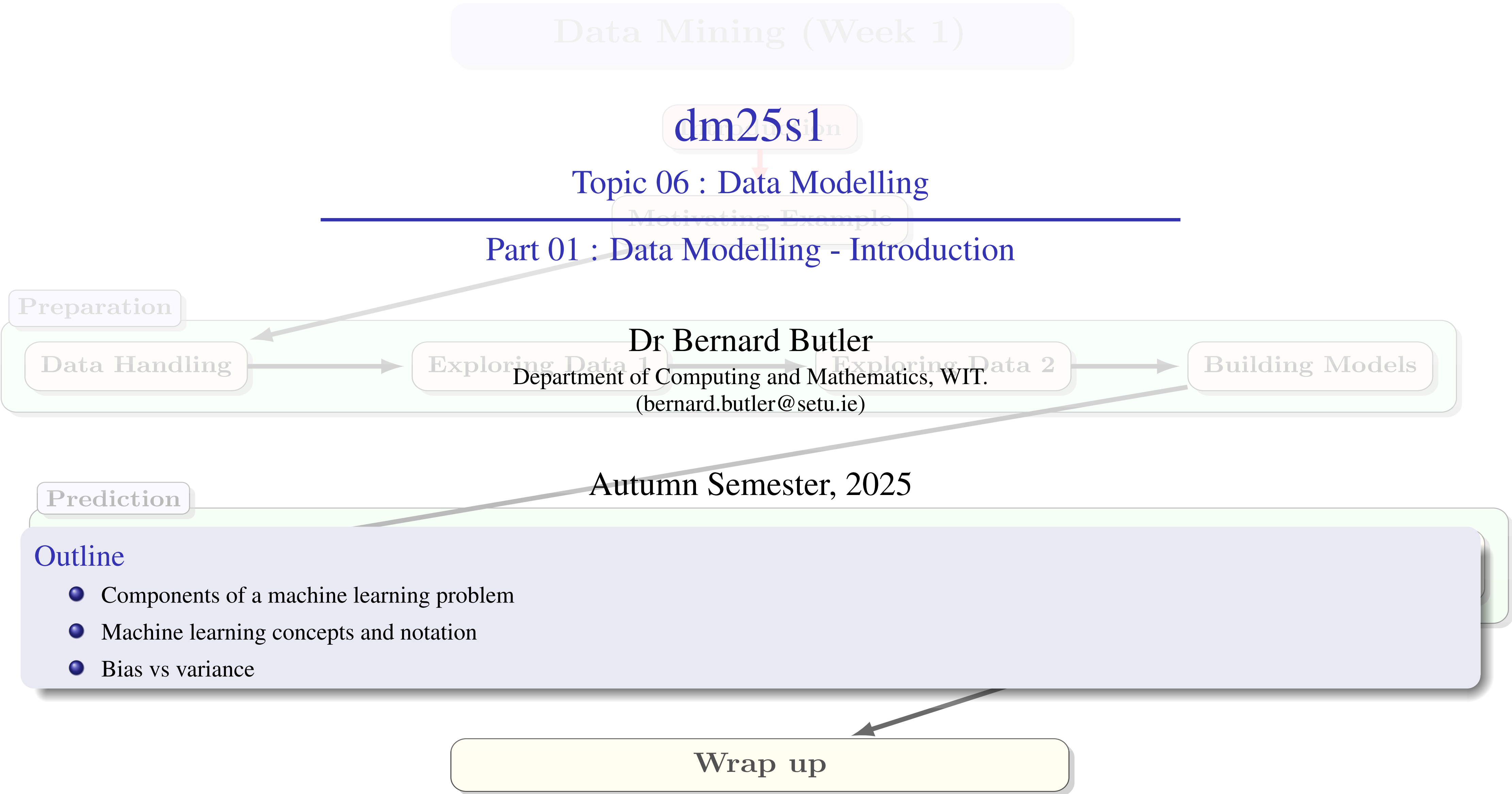
Department of Computing and Mathematics, WIT.

(bernard.butler@setu.ie)

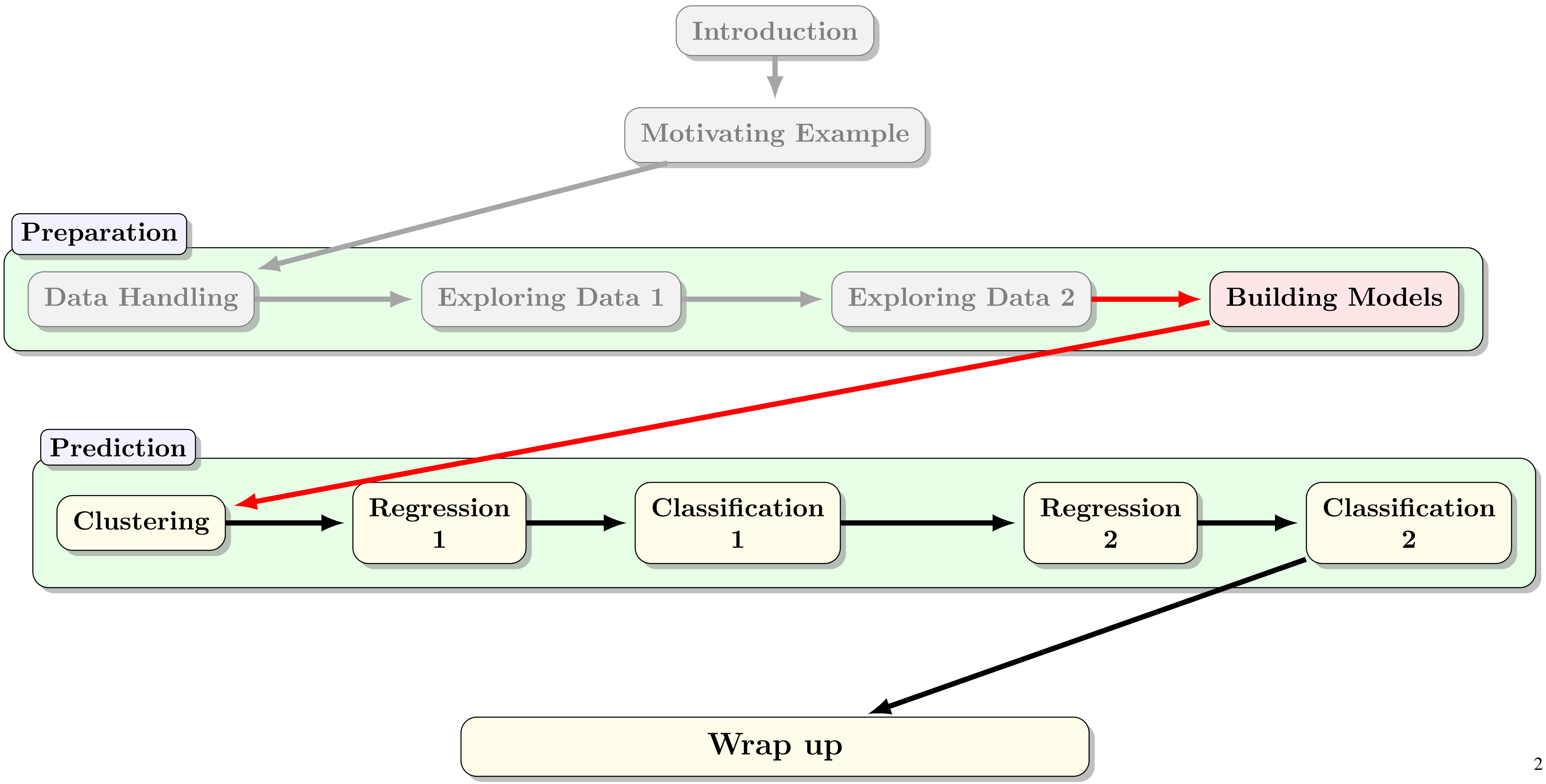
Autumn Semester, 2025

Outline

- Components of a machine learning problem
- Machine learning concepts and notation
- Bias vs variance



Data Mining (Week 6)



Terminology / Notation

$n+1$ columns / variables

$\boldsymbol{X}$

n features / attributes / dimensions

$\boldsymbol{y}$
target

m observations / instances / cases / rows

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0
2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	1
3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	1
5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0
6	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q	0
7	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S	0
8	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S	0
9	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S	1
10	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C	1
11	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S	1

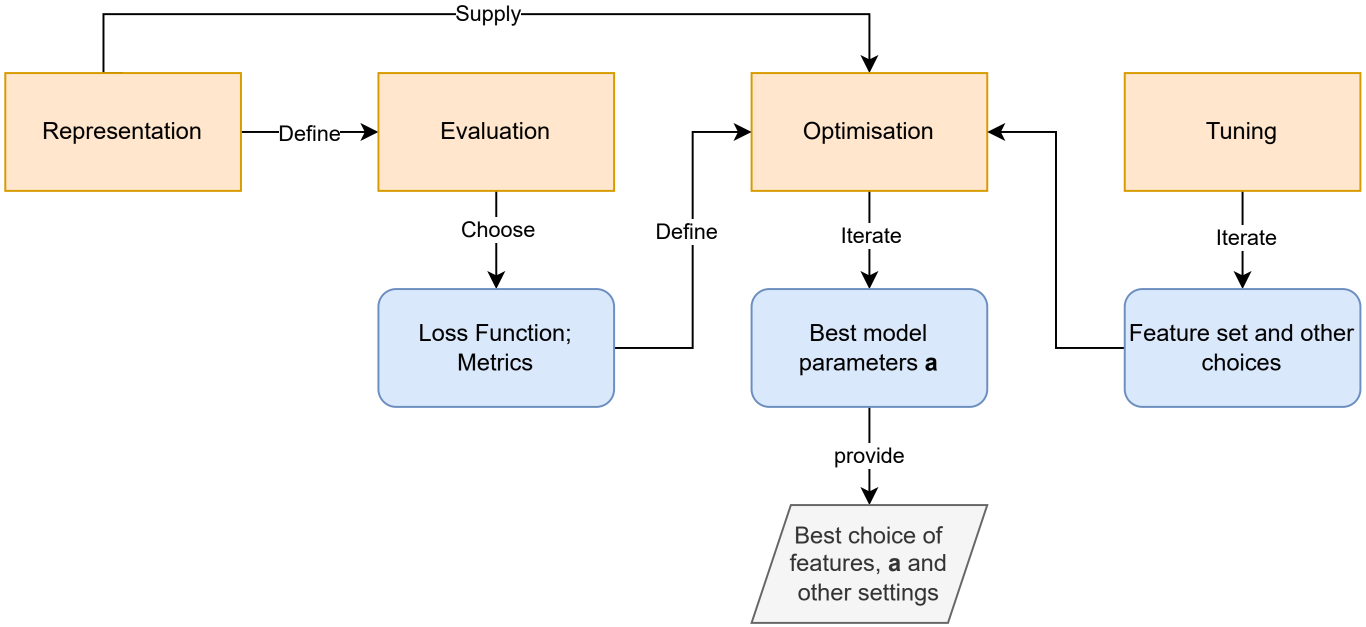
$\boldsymbol{x}^{(i)}$

$\boldsymbol{x}_j$

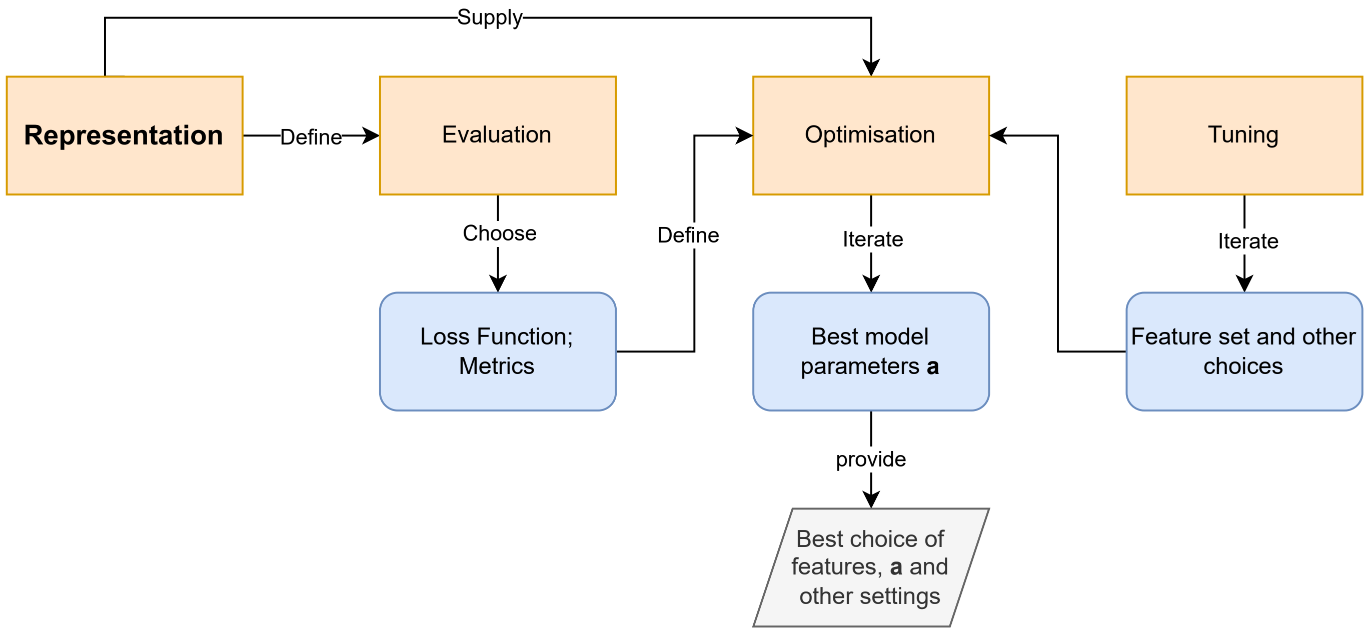
- A labeled dataset consists of m rows $\times$ $(n+1)$ columns / variables.
- Use bold to represent vectors and matrices.
- Use subscripts to indicate particular feature / attribute / column . . . . $\boldsymbol{x}_j$
- Use superscript in parenthesis to indicate particular observation / instance/ case / row . . . . $\boldsymbol{x}^{(i)}$
- So $x_j^{(i)}$ (or $x_{i,j}$) is the i-th observation in the j-th feature . . . . $x_j^{(i)}$

Components of a Machine Learning Problem

Where do we start????

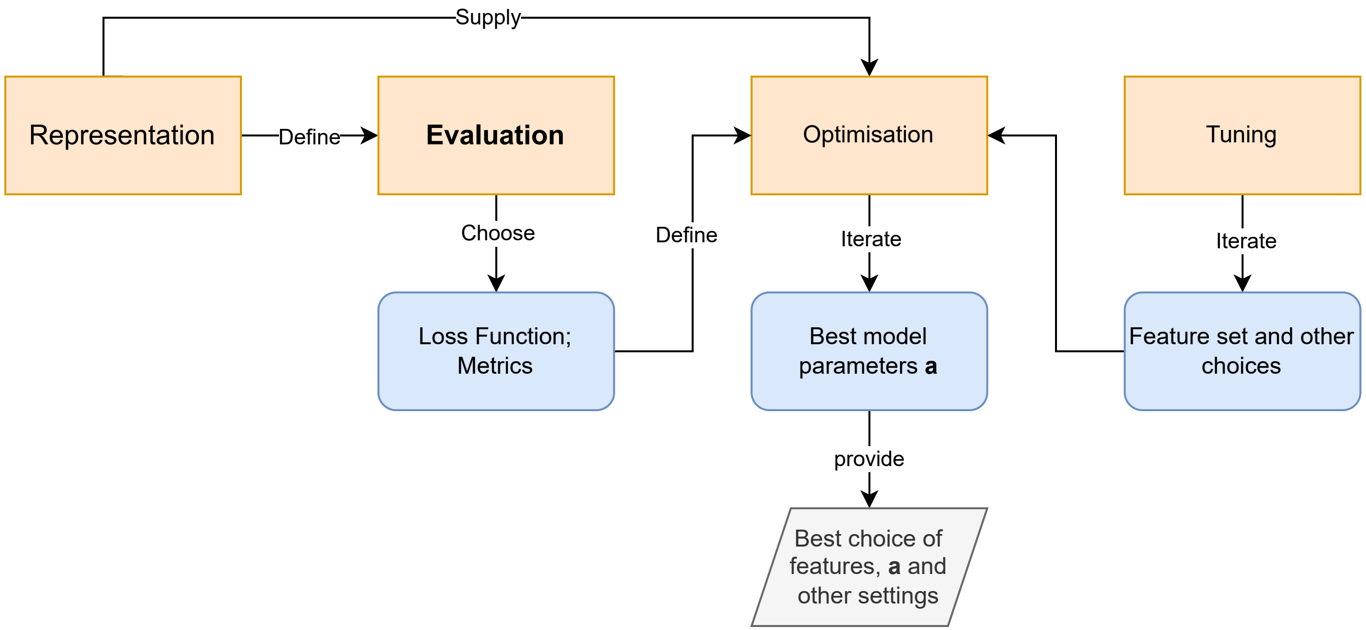


Component 1: Representation



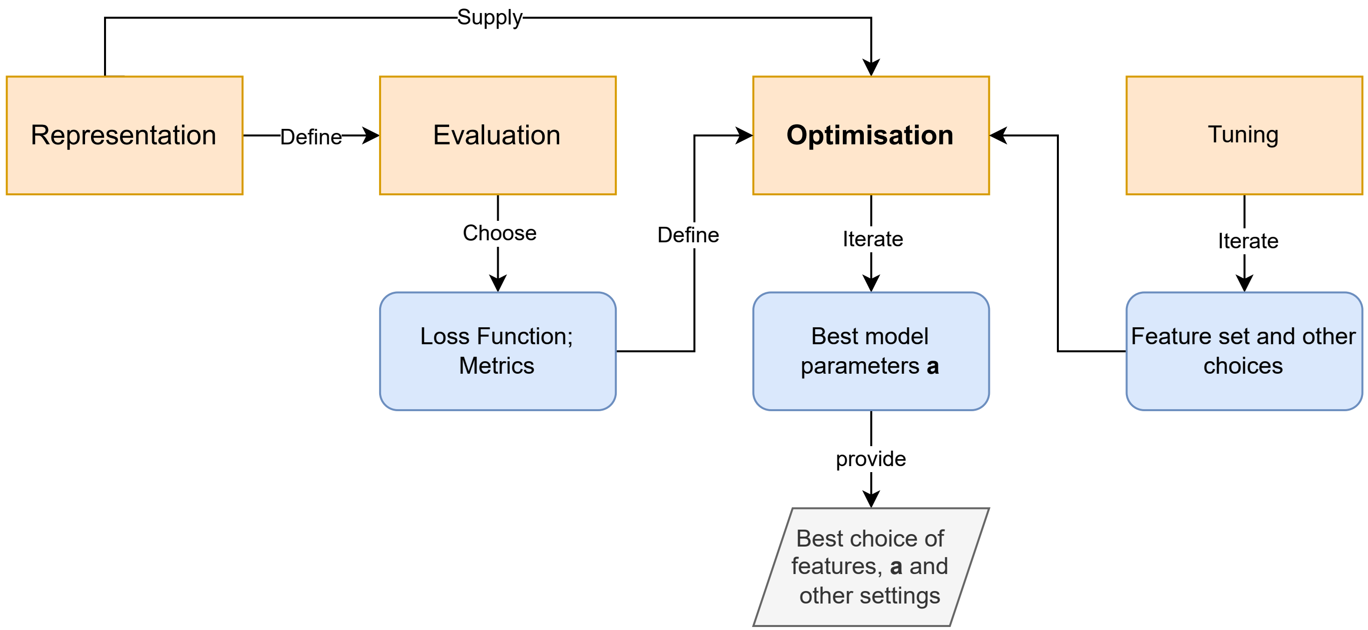
- Given the business objective and the data, choose what type of ML problem to solve: regression, classification, clustering, etc.
- Choose an algorithm for that problem. For example, if the ML problem is classification, we can choose KNN (other techniques will be seen later)
- What is the target and what features are available?
- These choices are made once - they are foundational for what follows.

Component 2: Evaluation



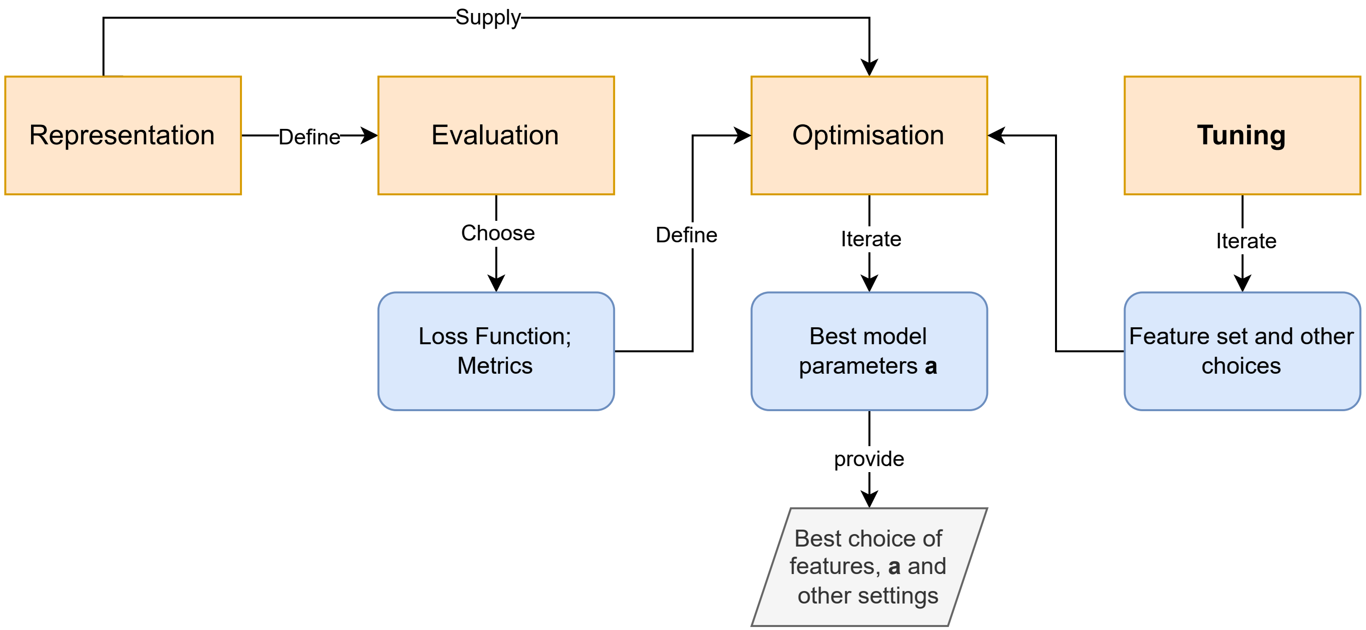
- Need an Objective Function that we can use to measure how well the model represents the data.
- For classification: how often does the model predict the right class, so that the predicted target and the actual target match each other?
- For regression: how close are the predicted target values to the actual target values?
- Generally the objective function takes actual and predicted target values and returns a single nonnegative number.

Component 3: Optimisation



- Most ML algorithms are *iterative*: start with a guess at the model parameters $\boldsymbol{a}$, and improve them until the process terminates.
- The nature of $\boldsymbol{a}$ and how the code searches for better $\boldsymbol{a}$ depends on the Representation, refined by the Tuning choices - we will see examples later
- The choice of when to terminate the optimisation procedure depends on the Evaluation choice.
- Most of the computational resources are needed for the Optimisation component.

Component 4: Tuning



- Tuning is also iterative and is applied on top of Optimisation
- Tuning parameters (also known as *hyperparameters*) are kept constant during Optimisation
- Tuning parameters depend on Representation. For example, we have seen one hyperparameter already: k in the KNN classifier.
- For Tuning, GridSearch (exhaustive over a range) and RandomSearch (pick represntative values) are popular ways to search for the best hyperparameter values.

Data Modelling (aka Machine Learning)

As alternative to the four component (Representation / Evaluation / Optimisationi/ Tuning) viewpoint we can think of a machine learning problem as

Definition 1 (Machine Learning)

Study of algorithms that improve their performance P at some task T with experience E.

Well defined learning task: $< P, T, E >$

- What metric should be used to measure performance?
- What cost function should be used?
- What is the cost of incorrect prediction?
- Computational cost?

- How complex is the task?
- Task type: classification, regression, . . .
- Linear vs nonlinear?
- What family of functions should be used?

- How many historical observations are needed?
- How accurate/noisy is the data?
- Do we have missing values?
- Is the data representative?

Taxonomy of Machine Learning Models . . .

. . . by Intuition/Motivation

- Geometric models use intuitions from geometry such as separating (hyper-)planes, linear transformations and distance metrics.
- Probabilistic models view learning as a process of reducing uncertainty, modelled by means of probability distributions.
- Logical models are defined in terms of easily interpretable logical expressions.

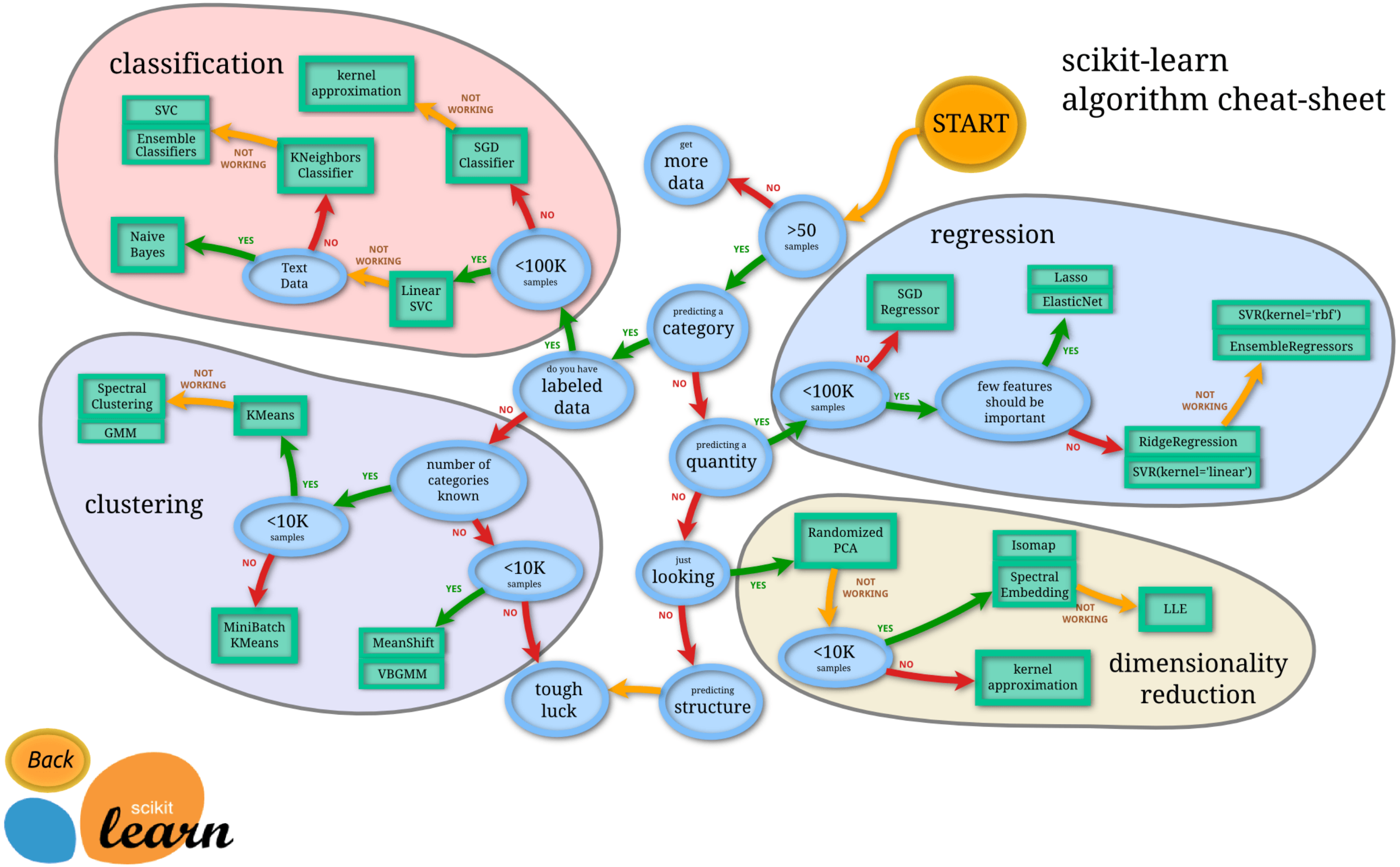
. . . by Algorithmic Properties

- Regression models predict a numeric output.
- Classification models predict a discrete class value.
- Neural networks learn based on a biological analogy
- Local models predict in the local region of a query instance.
- Tree-based models (recursively) partition the data to make predictions.
- Ensembles learn multiple models and combine their predictions.

. . . by Fixed/Variable Number of Parameters

- Parametric models have a fixed number of parameters.
- In non-parametric models the number of parameters grows with the amount of training data.

Aside: Scikit-learn Flowchart of Models (Shallow Learners)



A neural network with more than one hidden layer is called a deep learner, all other learners are shallow learners.

Statistical Models vs Machine Learning Models

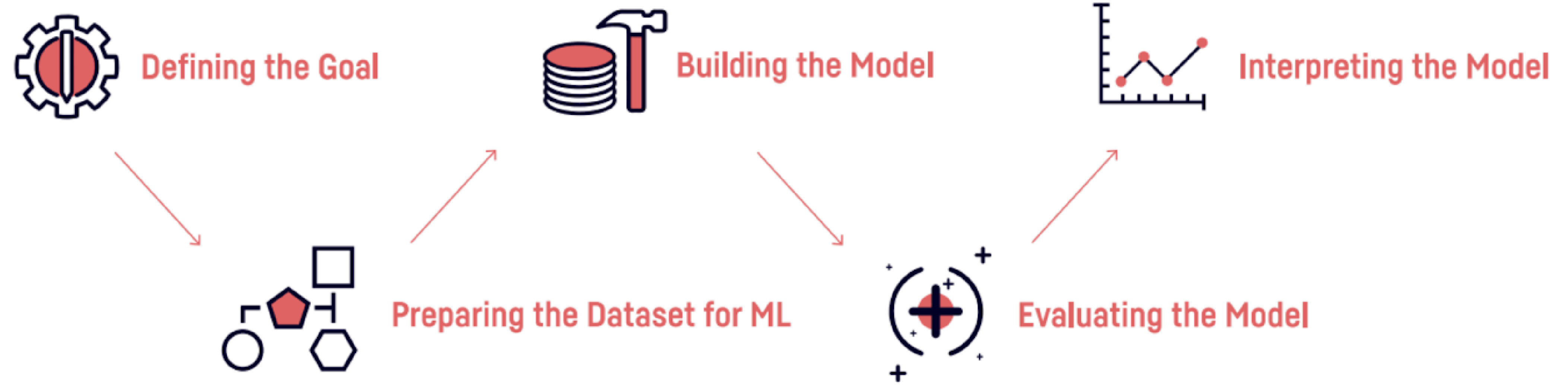
	Statistical Models	ML Models
Data	• Usually small ($<$ 1000 observations) • Low dimension ($<$ 10 variables) • Can have detailed understanding of data • Data is clean — human has looked at each data point	• Can be huge (million+ observations) • Large dimension (1000+, more for vision) • Too large for human to parse / understand • Data not clean — humans can't afford to understand/fix each point
Models	• Simple models — complexity limited by theory • Detailed/complex statistical assumptions re data • Model known, and data is carefully examined to verify assumptions.	• "No" upper limit on model complexity • Fewer statistical assumptions re data • Don't know right model? No problem! have multiple models and vote/weight results
Validation	• Evaluation based on theoretical estimates under stated statistical assumptions • Analysis of errors using theoretical distributions	• Empirical evaluation methods instead of theory — how well does it work on **unseen** data? • Don't calculate expected error, measure it from **unseen** data.

Statistics would be very different if it had been born after the computer instead of 100 years before

Splitting data into train+test(+validation) is vital

The Pipeline Metaphor

Model Building Pipeline



Source: Dataiku

Comments

- We saw the first two stages in previous weeks
- This week we look at the remaining stages
- Of course this pipeline is a simplification. In reality it is iterative.

What does a (supervised learning) model look like?

Definition 2 (Linear Model)

General form of linear model used in this module looks like

$$y_i \sim f_i^{(1)} + f_i^{(2)} + \ldots + f_i^{(\mathrm{n})}$$

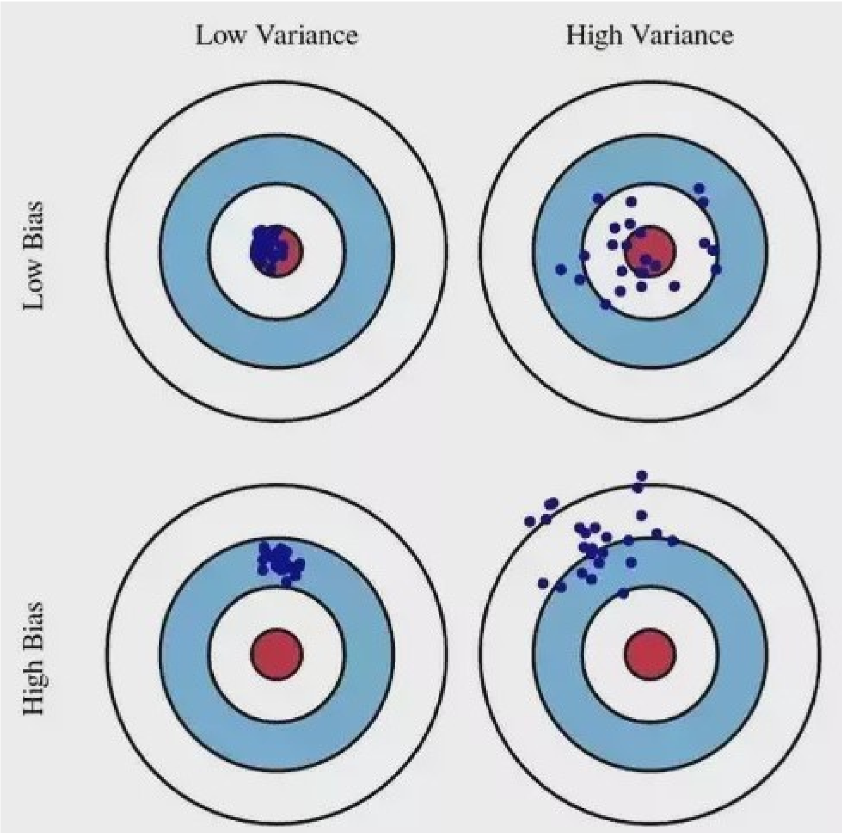
where y_i is the value of the response variable (target) for row (observation) i, and $f_i^{(\mathrm{j})}; j = 1, \ldots, n$ is the value of the j^{th} feature for that observation.
The model is linear in the sense that it can be turned into the following linear *equation*:

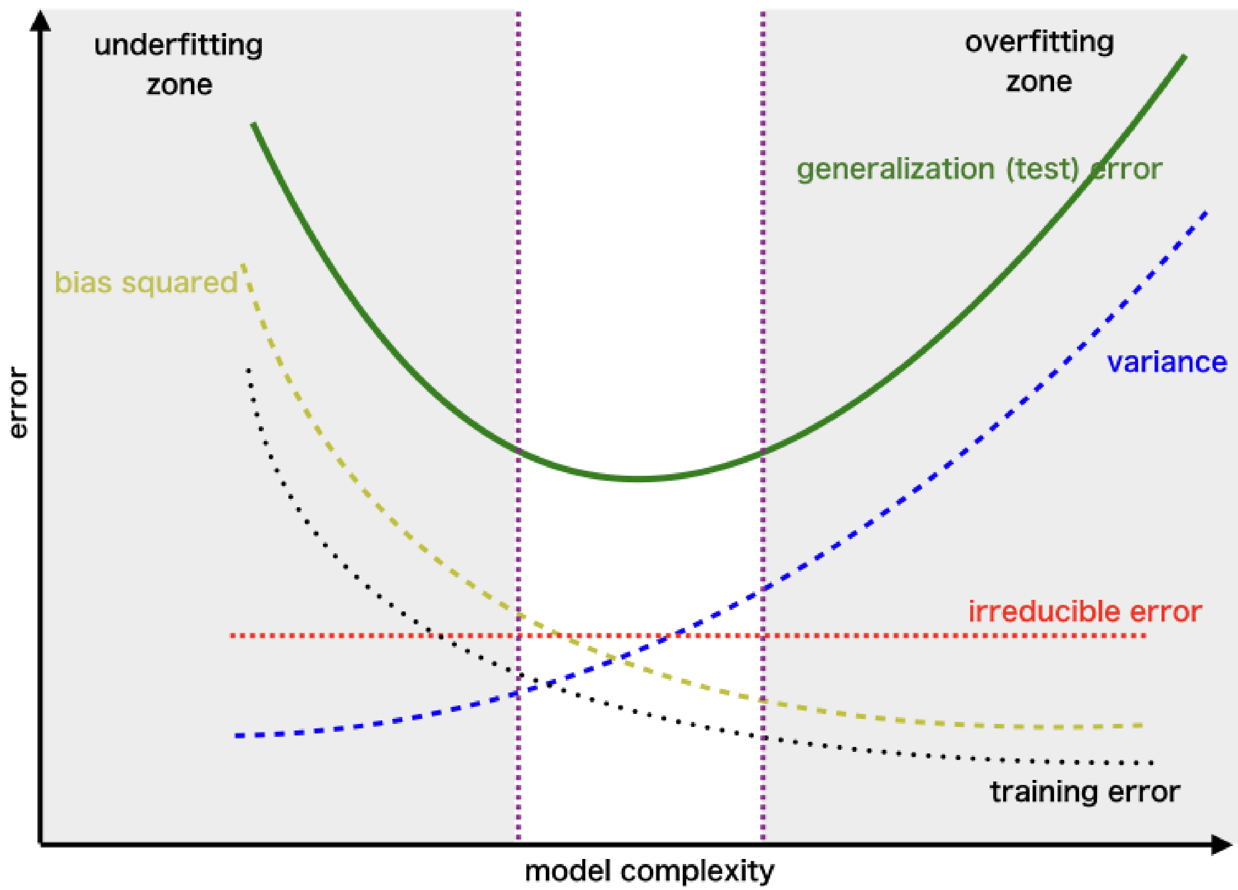
$$y_i = a_0 + a_1 f_i^{(1)} + a_2 f_i^{(2)} + \ldots + a_n f_i^{(\mathrm{n})} + \varepsilon_i$$

In words: each target value y_i in the data is modelled as a linear combination of the model parameters $\boldsymbol{a}$ and the features, plus some error.

- Note that the features f can be nonlinear (such as $\sin x$) but the model parameters $\boldsymbol{a}$ must appear linearly.
- The goal of modelling is to find $\boldsymbol{a}$ so that the *prediction error* (loss function $\sim \|\boldsymbol{\varepsilon}\|$) is a minimum.

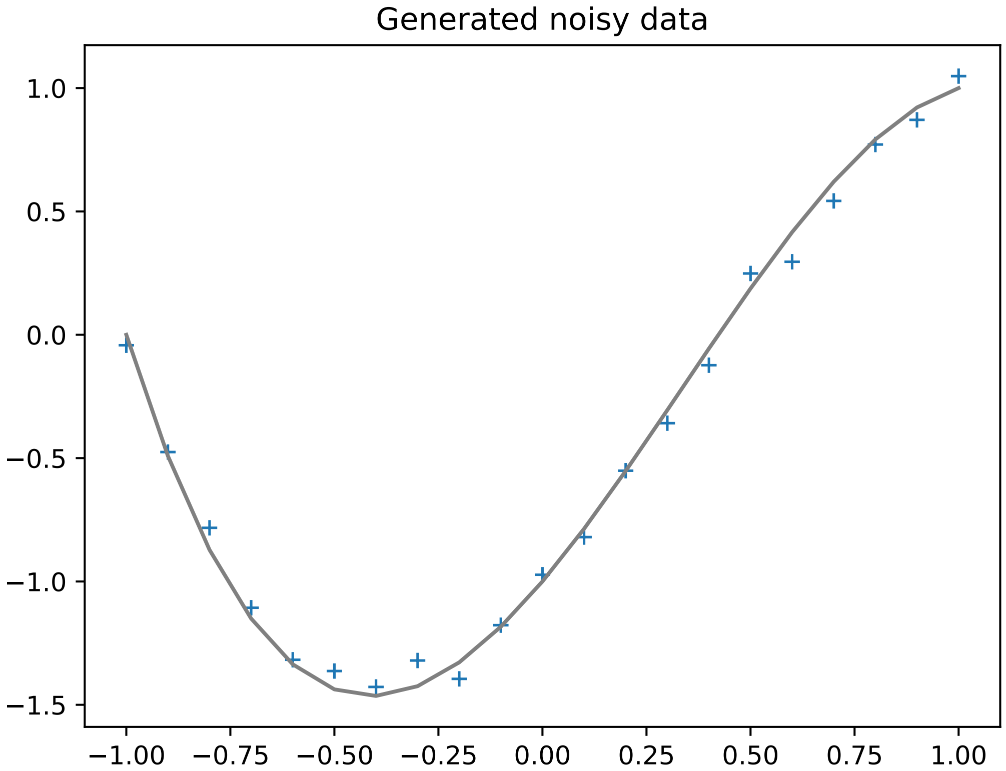
Bias-Variance and Total Error





Look for parameters $\boldsymbol{a}$ that minimise the generalization error (estimated using the test set that was not used during training)

Example: Noisy data

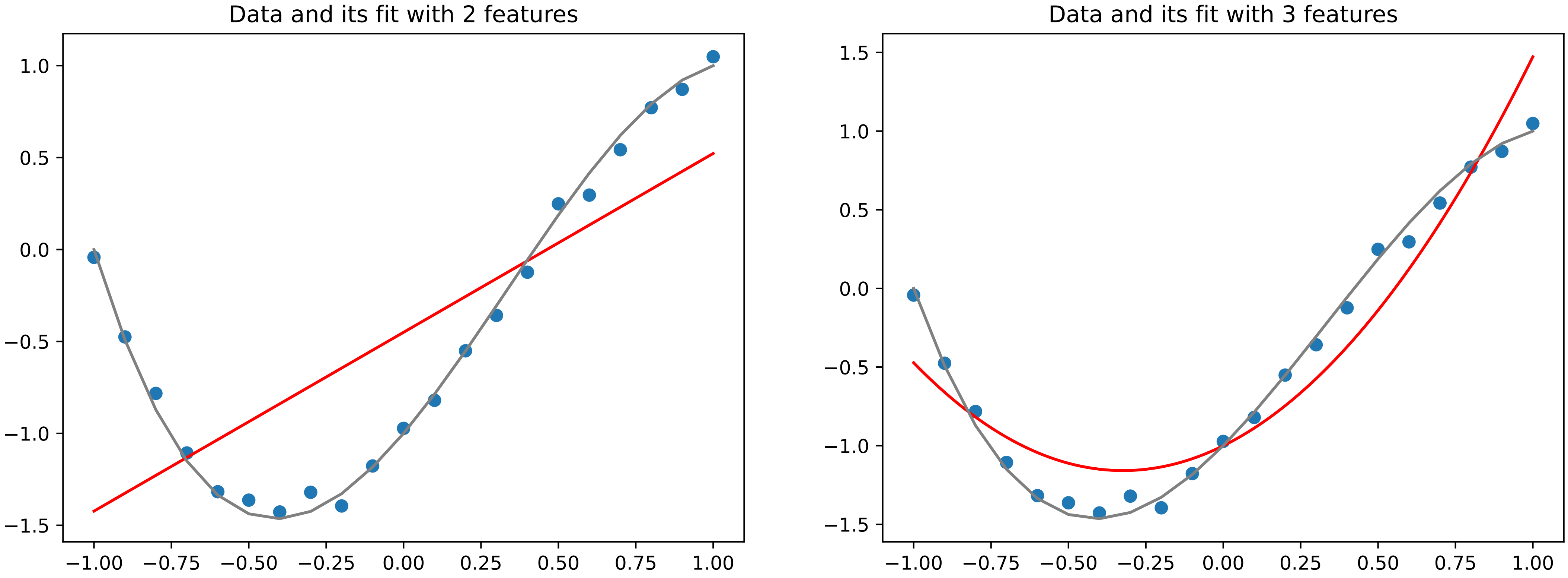


Comments

- Given data with some error (noise)
- Expected underlying model is indicated by the grey curve
- In the next slides we will compare different models, indicated by red curves
- The models have different numbers of *features*
- The values predicted by each model lie on the red curve
- The loss function is an estimate of how much the grey and red curves differ

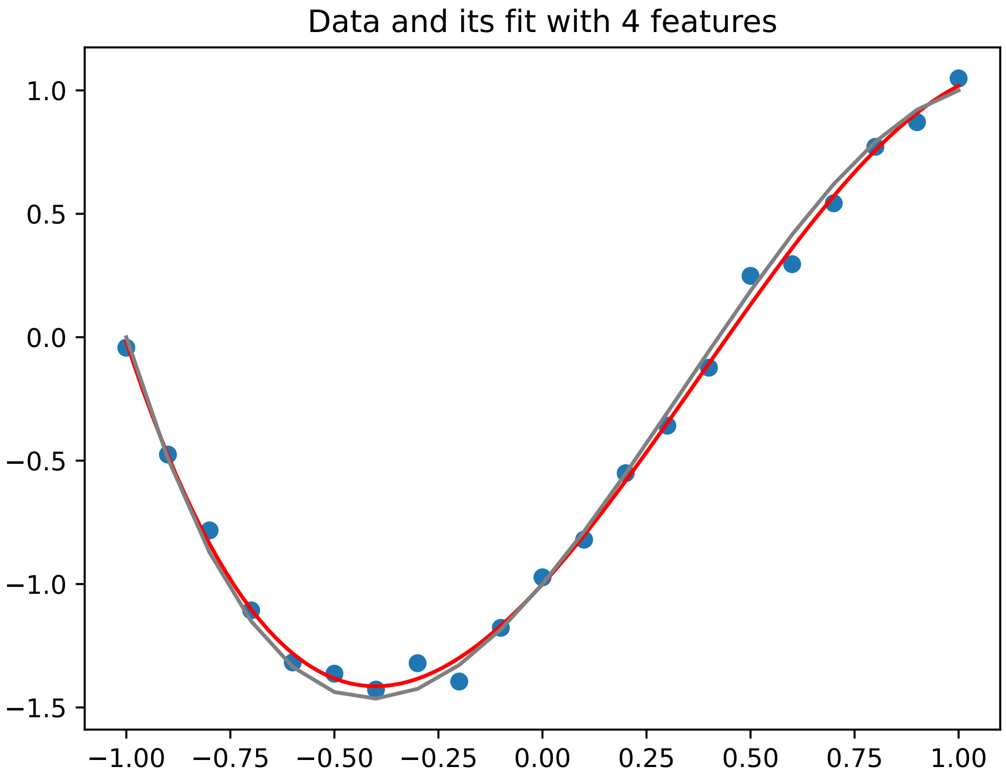
Look for the number of features that minimise the loss function

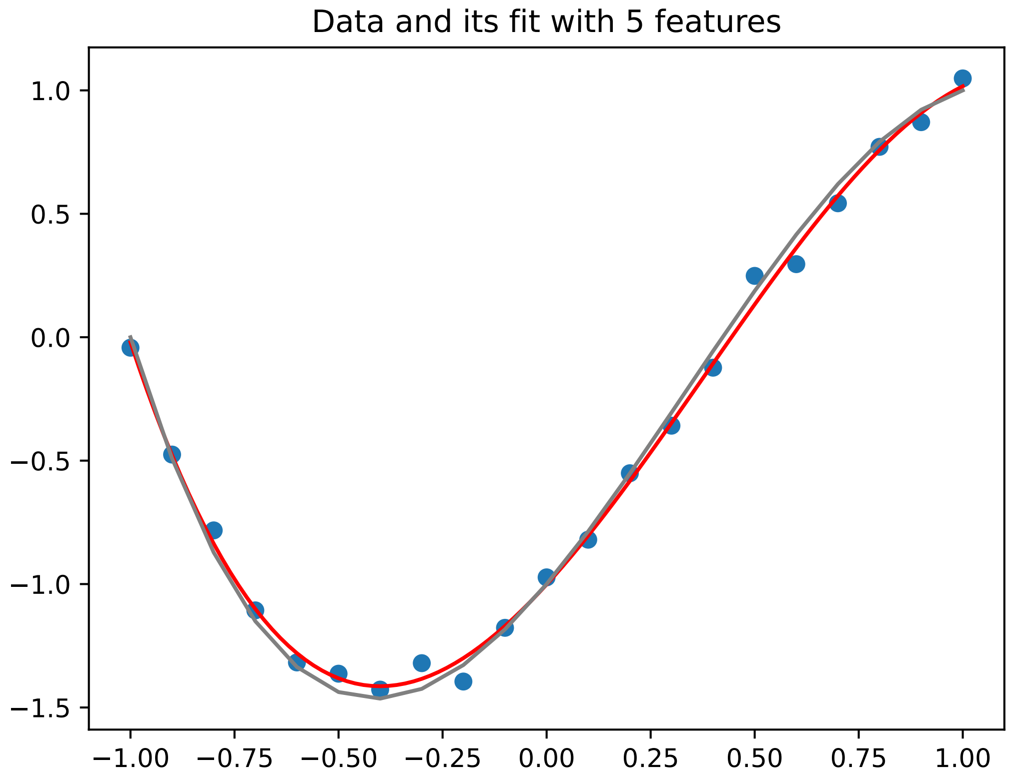
High Bias, Low variance



Need more features: add more - but which ones?. . .

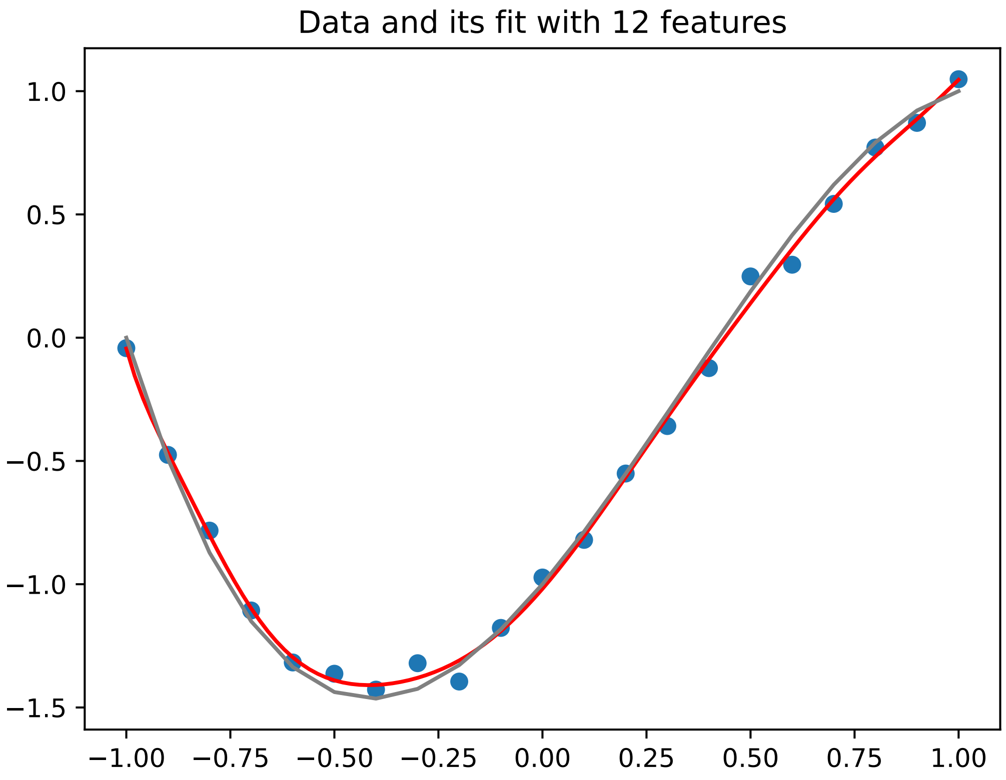
Low Bias, Low variance

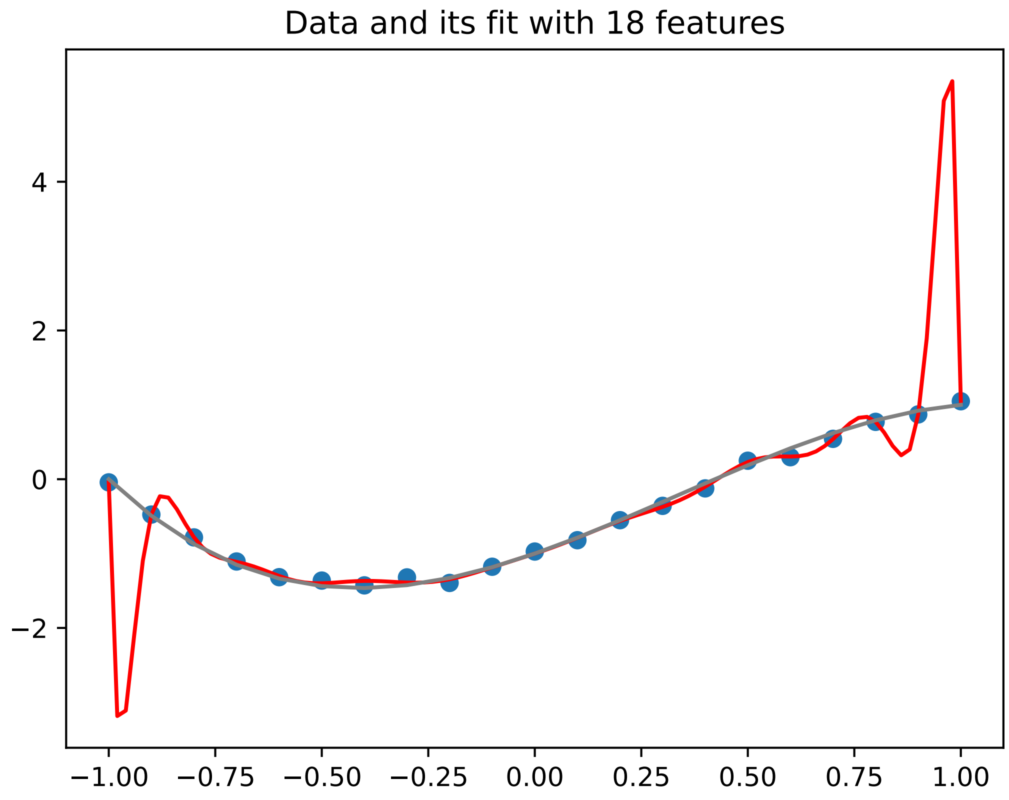




About the right number of features. . .

Low Bias, High variance





Too many features: remove some, but which ones?…

Example Model Types

Model	Applications	Concerns
Logistic Regression	X-ray classification	Regression with transformed variable
Fully connected networks	Classification	Classical ANN: choose encoding and size
Convolutional Neural Networks	Image processing	deep learning - choose segmentation
Recurrent Neural Networks	Voice recognition	ANN with feedback - how much?
Random Forest	Fraud Detection	Ensemble method - how many?
Reinforcement Learning	Learning by trial and error	Choose goal and penalties
Generative Models	Text, Image creation	Choose parameters
K-means	Segmentation	Choose distance function and k
k-Nearest Neighbors	Recommendation systems	Choose distance function and k
Bayesian Classifiers	Spam and noise filtering	Deal with imbalances

Before you start. . .

Does a *pre-trained* model exist?

Transfer Learning

- Building a model from scratch is resource-intensive
- Open source data and model exist, particularly for deep learning (not in this nmodule)
- Most frameworks provide example models that can be used as a template
  - Select a similar model
  - Prune it (remove unnecessary terms)
  - Train using the pruned model as a starting point