# Challenges in practical data science

Philip Jonathan

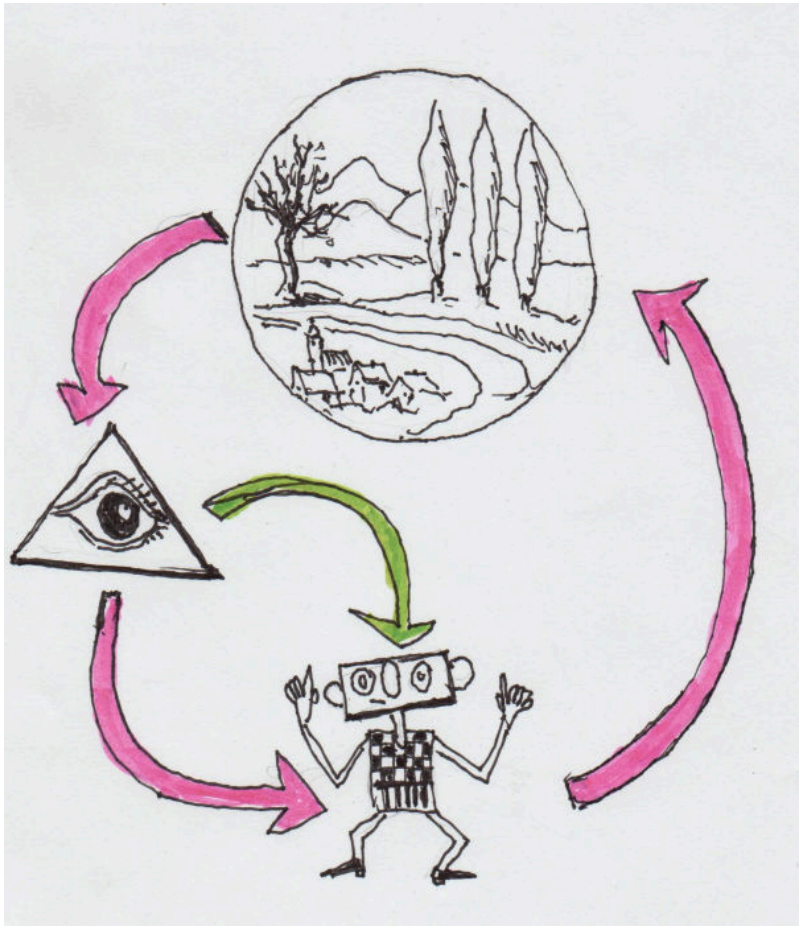
Lancaster University, Department of Mathematics & Statistics, UK.
Shell Research Ltd., London, UK.

Seminar, Data Science Institute
(slides at www.lancs.ac.uk/∼jonathan)


# Outline

- The digital transformation
- Representative applications
- Assessing empirical models
- Assurance in data science

- Acknowledgement
  - Colleagues in Shell
  - Academic partners



# Big picture

# Delivering a 'digital transformation'

- Digital roadmap
  - All parts of organisation involved (e.g. upstream, downstream, retail, finance, HR)
  - Convergence to common way of working
- Agile system development
  - Self-organising, cross-disciplinary teams
  - Common platform, governance, replication
  - Fail fast
- Delivery model
  - New technologies and R&D
  - Proof of concept studies, minimum viable products
  - Accelerators
  - Replication

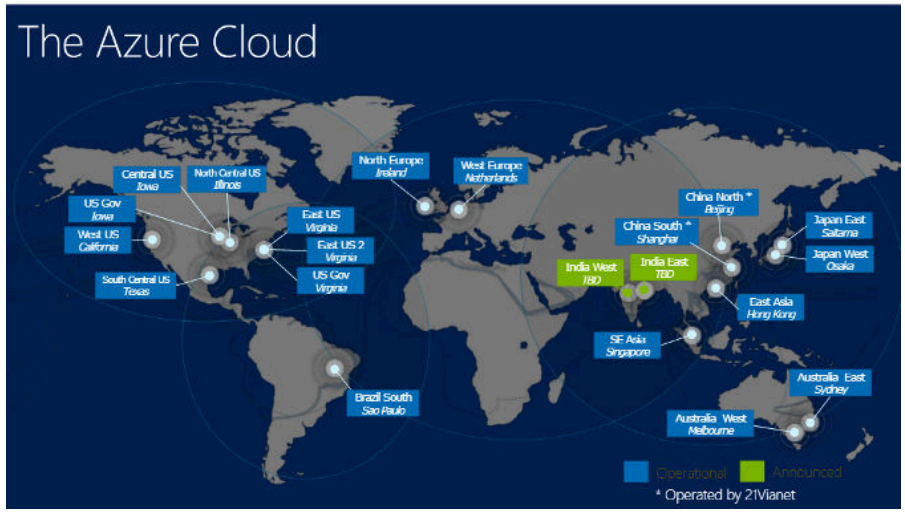
# What's changed?

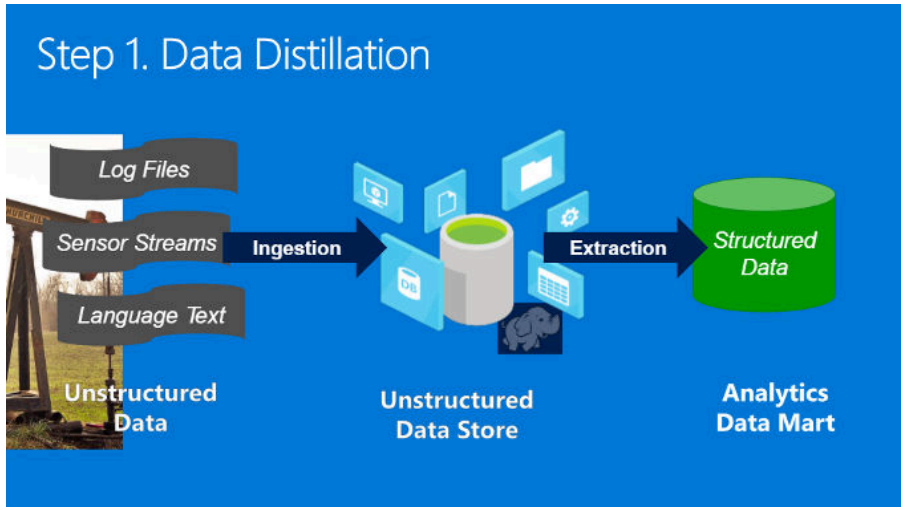
Basics

- Scale, speed, connectedness
- Numeric, text, image, sound
- Parallelism, cores, clusters, cloud
- Freeware (PYTHON)
- Data engineering systems

Cross-discipline

- Software engineering
- Computer science
- Data engineering
- Statistics & data science
- Communication

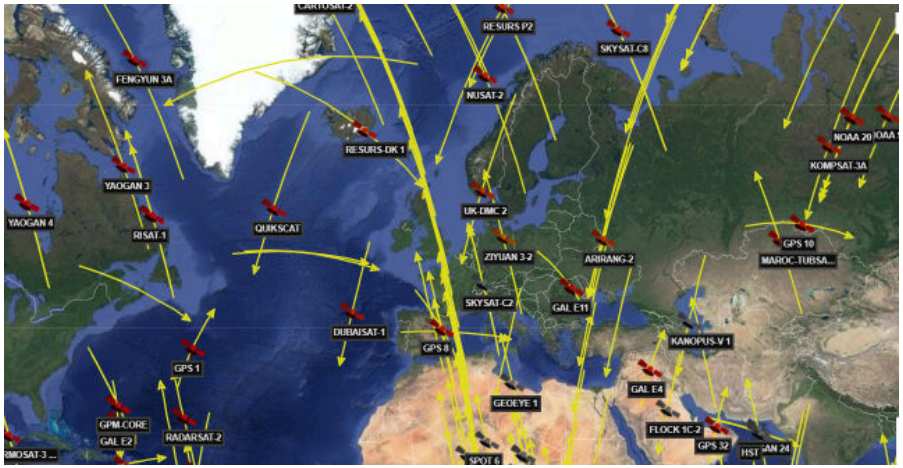


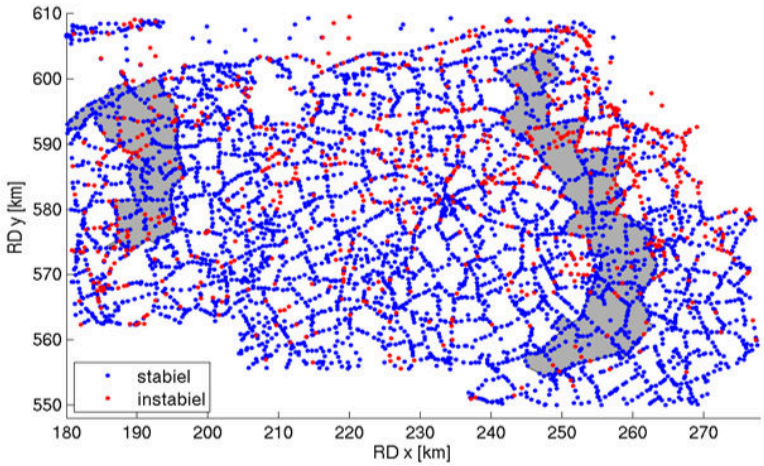



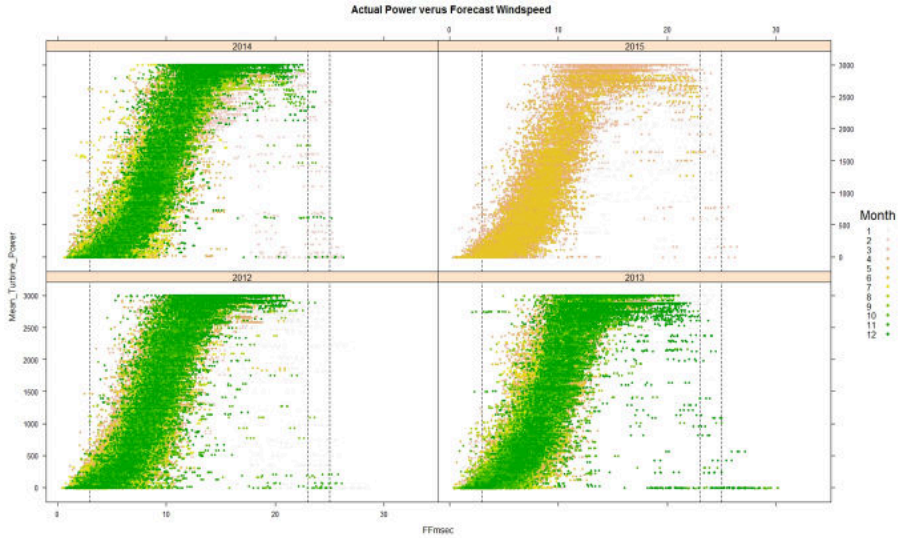

[Microsoft]

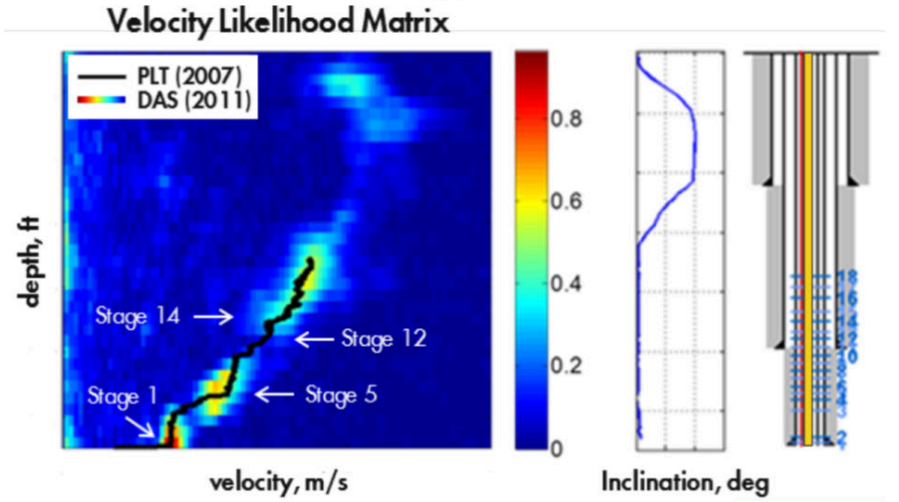
# Typical applications

# Physical environment

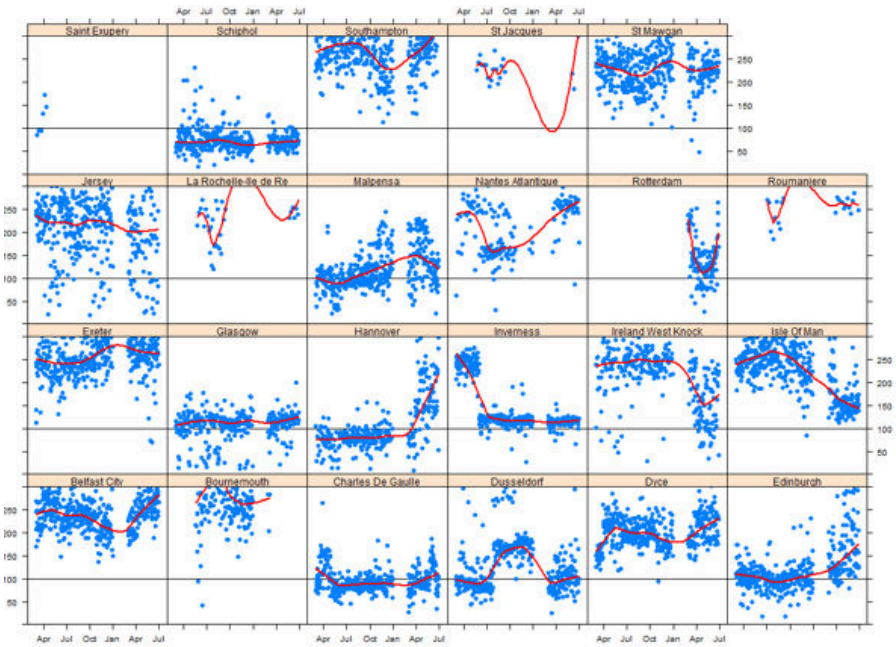


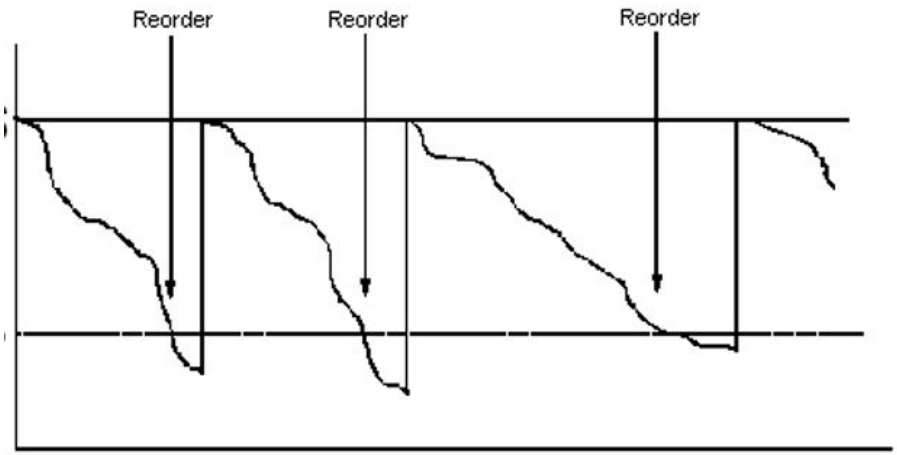



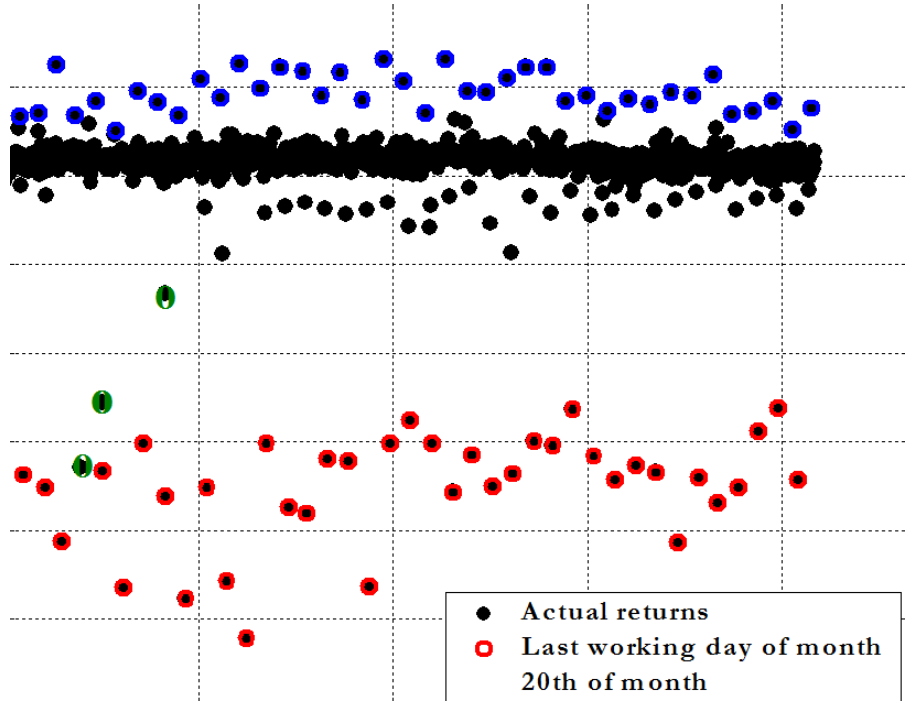



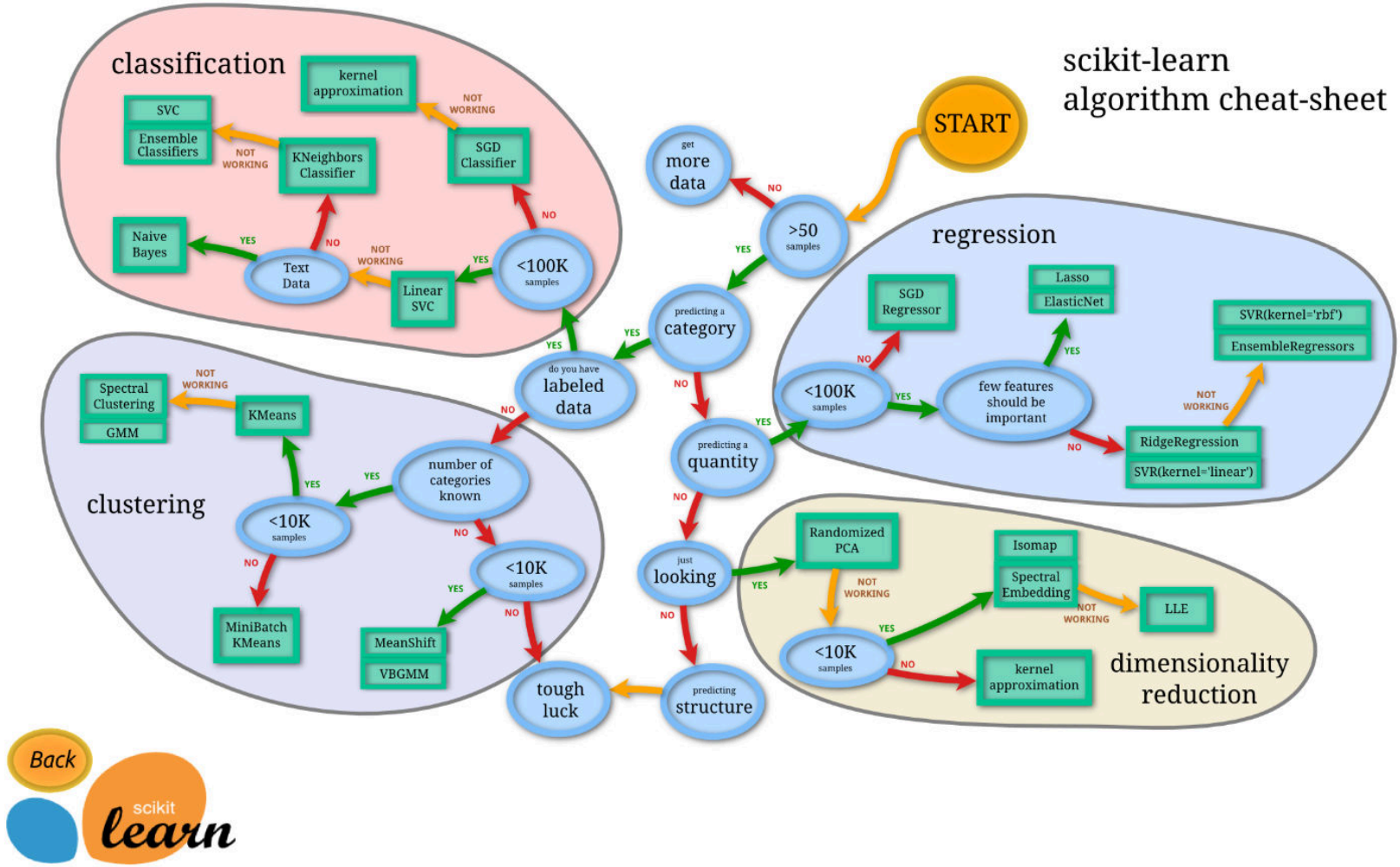
# Business, finance



# Standard toolkit




[sklearn]

# Typical process application

- Liquified natural gas (LNG)
  - Source gas → Liquid for transport → Gas for use
- World-wide facilities
  - Brunei, Oman, Nigeria, Australia, Qatar, Russia, Trinidad&Tobago, Egypt
- Nigerian plant
  - Six processing units
  - Total processing capacity: 22 million tonnes of LNG p.a.
  - Accounted for approximately 7% of global LNG supply in 2017

(Information from `www.shell.com`)

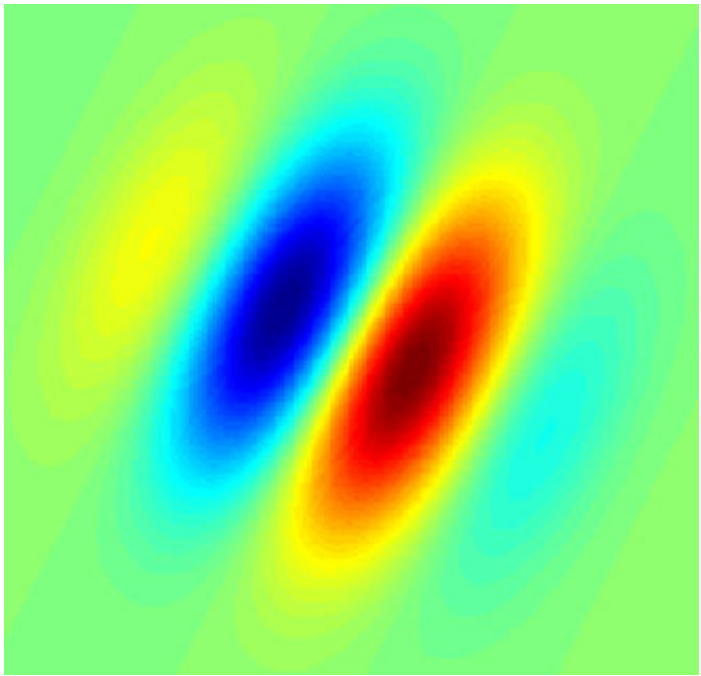
# Typical process application

- System: input $\rightarrow$ process $\rightarrow$ output
  - Flows, Temperatures, Pressures, Compositions, Valve settings, Level settings
  - Recycles, automatic control, constraints
- Data
  - Multivariate time-series, $\log_{10}(p) \in (2, 4)$, $\boldsymbol{X}(t), \boldsymbol{Y}(t), t \in \mathcal{T}$
  - Some of $\boldsymbol{X}$ manipulable ($\boldsymbol{X}'$), others not ($\boldsymbol{X}''$)
  - $\mathcal{T} \approx 5$ years, sampling $\approx 1$ Hz, $\log_{10}(n) \in (3, 8)$
- Goal: optimisation
  - $f_1(\boldsymbol{X}'_{\mathcal{T}}, \boldsymbol{X}''_{\mathcal{T}}) = 0$, $\boldsymbol{Y}_{\mathcal{T}} = f_2(\boldsymbol{X}'_{\mathcal{T}}, \boldsymbol{X}''_{\mathcal{T}})$, $\boldsymbol{X}''_{\mathcal{T}} = f_3(\boldsymbol{X}'_{\mathcal{T}})$
- Issues:
  - Data quality (redundancy, reconciliation)
  - Complex time-series dependence structure (models are 'static')
  - Existing models at play (APC, engineering: $f = f_{\text{ENG}} + f_{\text{Emp}}$)
  - In-situ performance
- Most popular models (linear & regularised regression)

# Computer vision, NLP

- Raw input (images, text, sound) $X_0$
- Multiple filters $X = F(X_0, \theta)$
- Inference on processed $X$
- Optimal choice of filter (kernels) for prediction

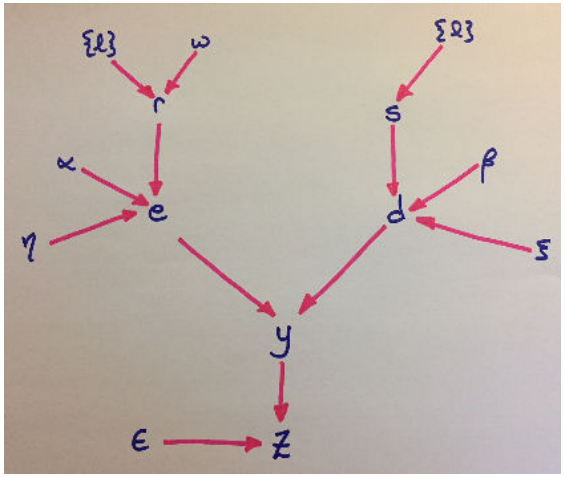
- Sparse + low rank
- Higher criticism (Donoho)



Gabor filter [wiki]

(More illustrations at www.shell.ai)

# Uncertainty quantification, emulation



A simple system model

- Flexible framework, Bayes linear
- Optimal design
- Probabilistic ODEs, Bayesian optimisation

$$
\begin{aligned}
\text{Obs} &: z(x) = y(x) + \epsilon \\
\text{Sys} &: y(x) = e(x) + d(x) \\
\text{Emul} &: e(x) = \alpha' g(x) + r(x, \omega) + \eta \\
\text{Disc} &: d(x) = \beta' h(x) + s(x) + \xi
\end{aligned}
$$

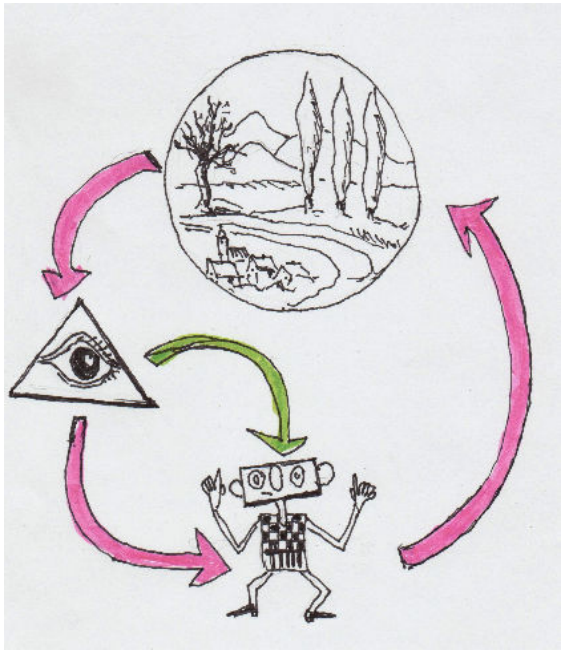
$$
\begin{aligned}
e &: \text{emulator or 'process' model} \\
d &: \text{discrepancy model} \\
g, h &: \text{bases for covariate space} \\
r, s &: \text{Gaussian process residuals}
\end{aligned}
$$

$$
\begin{aligned}
\text{Priors} &: \text{all Gaussian} \\
\text{Data} &: \text{emulator } E, \text{measured } Z \\
\text{Estimation} &: f(\alpha, \beta, \{\ell_r\}, \{\ell_s\}, \omega | E, Z) \\
\text{Prediction} &: f(y(x) | E, Z)
\end{aligned}
$$

# Robotics, automatic decisions

Markov decision = 'Reinforcement learning'

- 'Agent' observes state $s_t$
- Takes action $a_t$
- Leads to state $s_{t+1}$ with $\mathbb{P}\ p(s_{t+1}|s_t, a_t)$
- Agent receives reward $r_t = r(s_{t+1}, a_t, s_t)$
- Policy $\pi(a|s)$ governs behaviour
- History $h = [s_1, a_2, s_2, a_2, ..., s_T, a_T]$
- Discounted return $R(h) = \sum_t \gamma^t r_t, \gamma \in [0, 1)$
- $p_\pi(h) = [\prod_t p(s_{t+1}|s_t, a_t)\pi(a_t|s_t)]\ p(s_1)$
- Optimal $\pi^* = \underset{\pi}{\text{argmax}}\ \mathbb{E}_{p_\pi(h)} R(h)$



- Boston Dynamics (dogs opening door, robot somersault; YouTube)

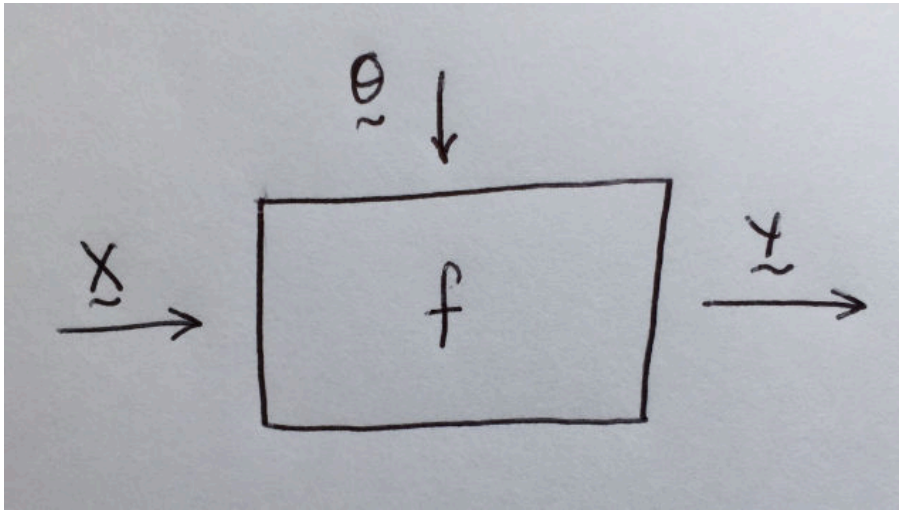
# Assessing empirical models

# Fundamentals

- Think stand-alone empirical modelling (not robotics)
- Data not from a designed experiment
  - Interpolation (failure to fit)
  - Extrapolation (failure to generalise)
  - Curse of dimensionality: often can't tell one from the other
- Data preparation and preprocessing
- Naive presumptions
  - IID, ignore dependence structure, time-series!
- Use cases
  - Non-expert users
  - Automatic, embedded predictions

# Assessing predictive algorithms

- $\boldsymbol{Y} = f(\boldsymbol{X}, \boldsymbol{\theta})$
- Black-box
  - Manipulate inputs, $\boldsymbol{X}$
  - Manipulate some tuning parameters, $\boldsymbol{\theta}$
- Predictive performance
- Prediction uncertainty
- Exceptionality



# In-vitro performance

- Competitions
  - e.g. Makridakis (M3, M4) forecasting, Kaggle, NN3, imagenet
  - e.g. Fernandez-Delgado ($\sim$ 200 classifiers), OpenML ($\sim$ 100 classifiers)
  - e.g. MLaut regression and classification (Kiraly), AutoML (Freiburg)
- Findings for time-series (M3) and general prediction (Kiraly): ensemble methods best, GPs/SVMs close, ML (inc. 'deep learning') poor
- Lazy focus on 'predictive $R^2$' not prediction uncertainty

**Table 10. Features of various Artificial Intelligence (AI) applications.**

| Type of Application | Rules are known and do not change | The environment is known and stable | Predictions can influence the future | Extent of Uncertainty (or amount of noise) | Examples |
|---|---|---|---|---|---|
| Games | Yes | Yes | No | None | Chess, GO |
| Image and speech recognition | Yes | Yes | No | Minimal (can be minimized by big data) | Face Recognition, Siri, Cortana, Google AI |
| Predictions based on the Law of large numbers | Yes | Yes | Minimally | Measurable (Normally distributed) | Forecasting the sales of beer, coffee, soft drinks, weather etc. |
| Autonomous Functions | Yes | Yes | No | Can be assessed and minimized | Self-Driving Vehicles |
| Strategy, Competition, Investments | No | No | Yes, often to a great extent | Cannot be measured (fat tails) | Decisions, Anticipations, Forecasts |
| Combinations of the above | It may be the ultimate challenge moving towards GAI (General AI) but also increasing the level of complexity and sophistication of algorithms | | | | Eventually it can cover everything |

# In-vivo performance

- Interpretability
  - Functional form for $\hat{f}$
  - Variable importance (easy for regression, trees; difficult for others)
- Parsimony
  - Variable selection not regularisation
- 'Human in the loop'
  - 'What legal scholars should learn about machine learning' (Lehr, Ohm)
  - Algorithms making decisions with no human supervision
  - Business- and safety-critical
  - Reinforcement learning of 'human experience' (Faisal)
- High level architectures enable but constrain
  - e.g. Azure DataBricks (database)
  - e.g. TensorFlow (images)
- Solution maintenance
  - Version control
  - Documentation

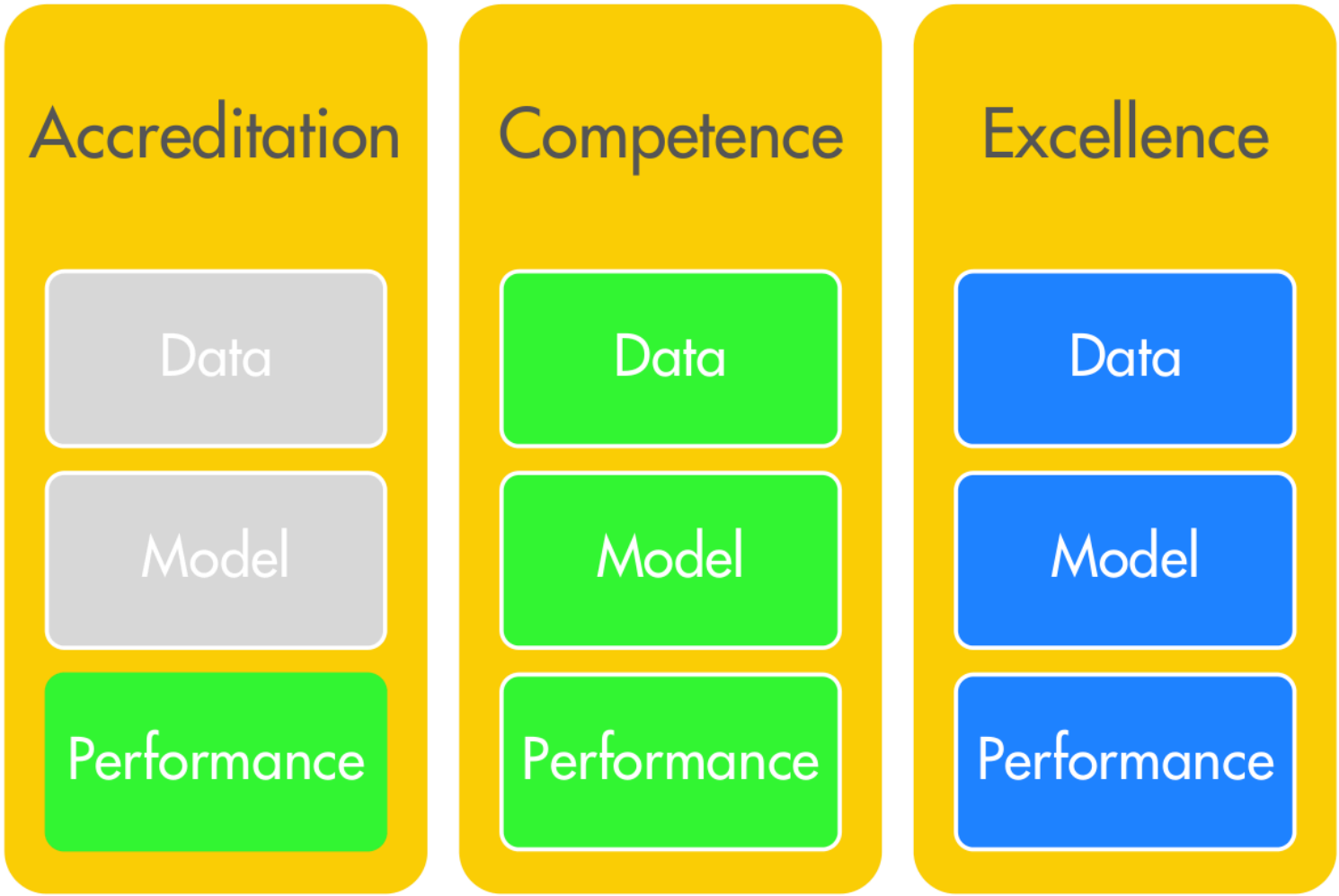
# Performance, uncertainty, exceptionality

- Predictive performance
  - Cross-validation, $\hat{\theta} = \text{argmin}_\theta L(\theta)$
  - Preserve dependence, choice of partition, bias, nested
  - Problem : does not provide $\text{var}(\hat{\theta}) \Rightarrow$ use BS also
- Prediction uncertainty
  - Bootstrapping (BS), $p_{\text{BS}}(\hat{\theta})$
  - Preservation of dependence structure
  - Problem : not inherently out-of-bag $\Rightarrow$ use CV also
- Exceptionality
  - Randomised permutation (RP) testing
  - $E = \mathbb{P}_{p_{\text{RP}}}\left[L(\hat{\theta}) \leq L(\hat{\theta}_{\text{RP}})\right]$
- Model-agnostic assessment ('wrapper')

- c.f. Bayes: posterior, evidence, posterior predictive, estimate or minimise generalisation loss
- Huge model-specific applied literature
- Conformal inference (CMU)

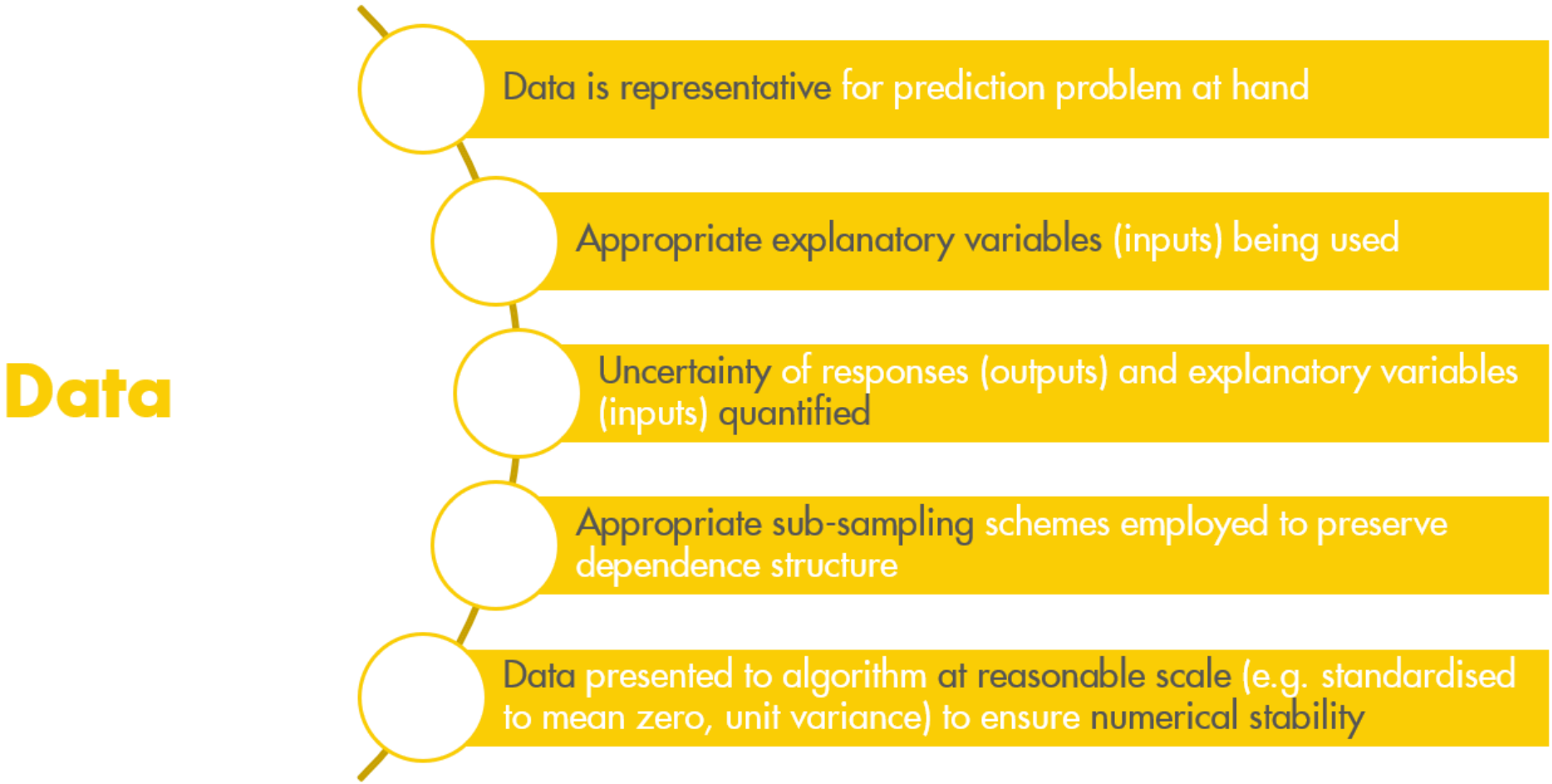
# Assurance in data science

# Fundamentals



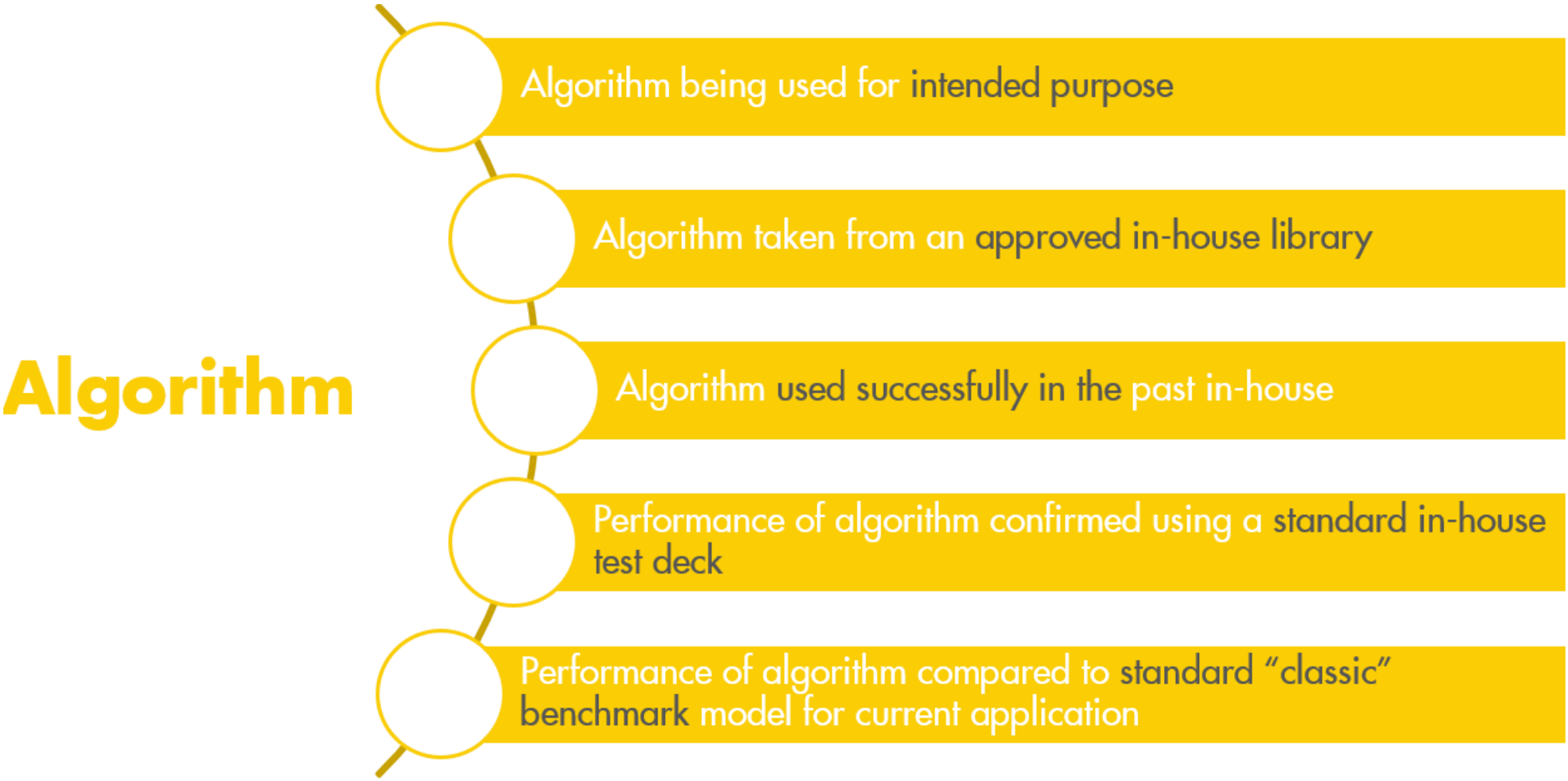

Accreditation, Competence, Excellence

# Data



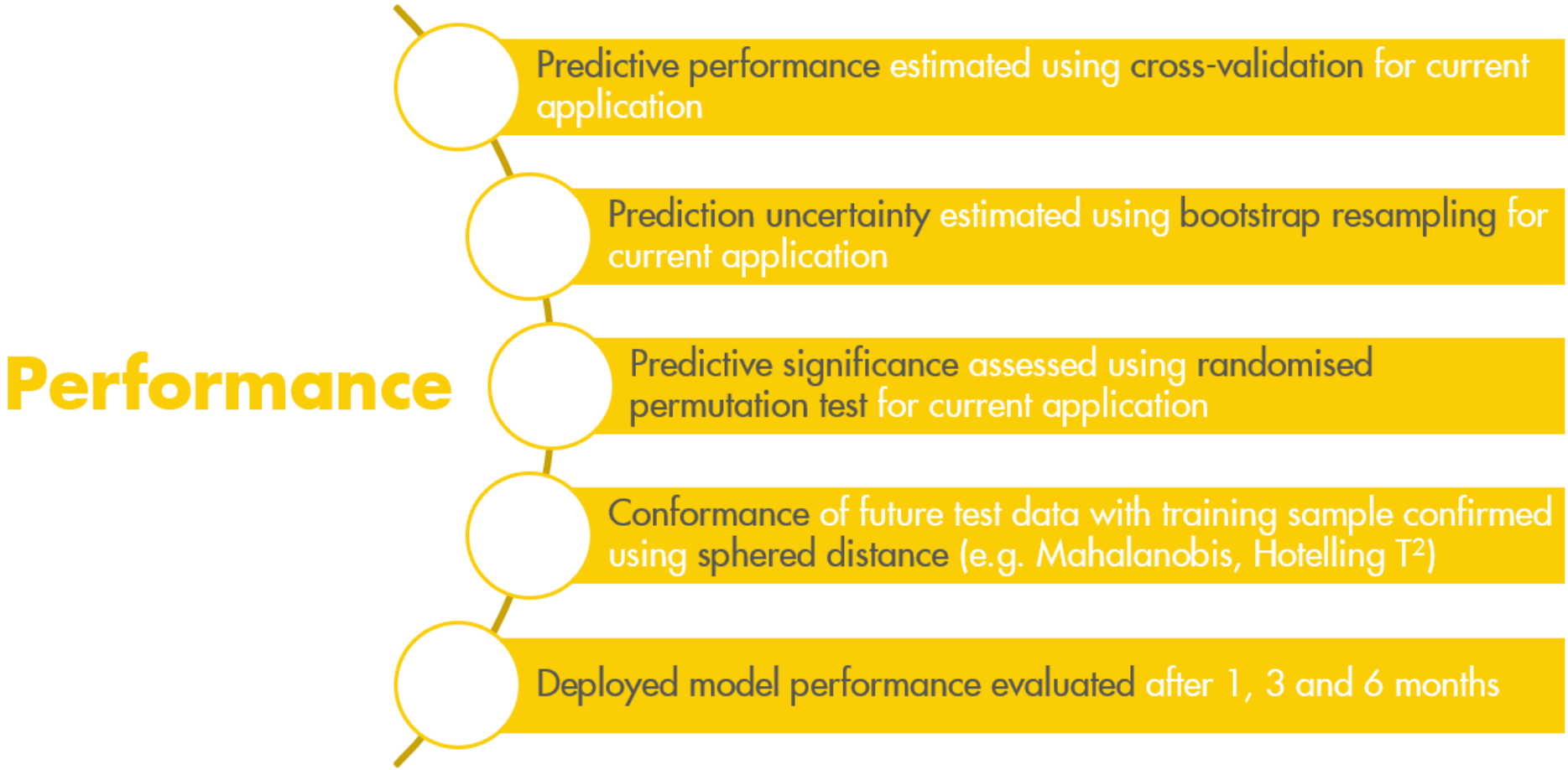

Assessing the data

# Algorithm




Assessing the algorithm

# Performance




Assessing performance

# Long-term

- Good governance
- Standardisation
  - Approved algorithms
  - Test decks
  - Reporting framework
- Continuous improvement
  - Monitoring of applications in development
  - Monitoring of deployments
- Learning
  - Data scientists
  - Expert reviewers
- Impact from data science

# Reasons for concern, optimism

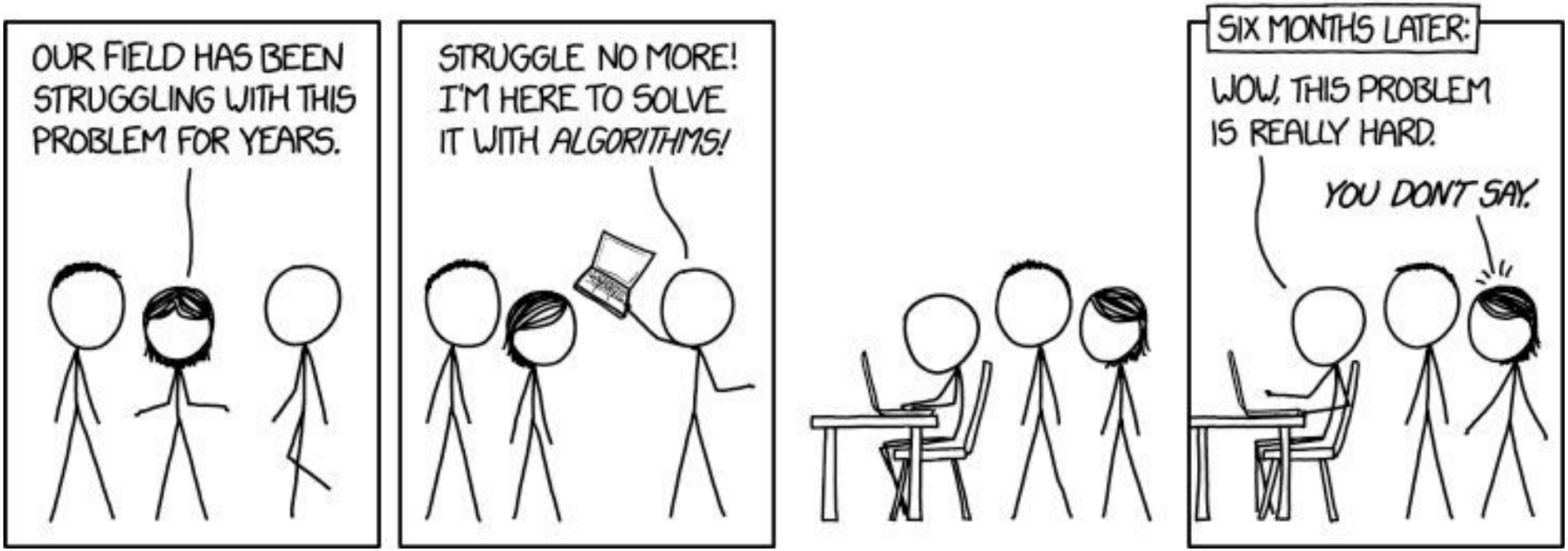
# Concern

- Data-driven models everywhere
  - Don't collect data until you know why you're collecting it
- Bias and uncertainty
- 'Lazy' application, just because it's possible
  - Current criticism of 5G
- Intrusive application
  - Spam emails, phone-calls, inadvertent discrimination
  - Snooping by third parties, phishing, cameras, voice monitoring
  - Terrorism, autonomous weapons
- Erroneous application
  - Boeing 737Max8
  - Autonomous vehicles
- Uncontrolled application
  - 'Black Mirror', AI / AGI takes over (Berners-Lee on www: 'downward plunge to dysfunctional future')
  - Nascent ethical, legal, governmental frameworks
  - Moral choices, dilemmas; intelligence vs. wisdom

# Optimism

- Data-driven models everywhere
  - Broad range of application complexity
- Bias and uncertainty
  - What statisticians and data scientists know how to do well
- Game-changing
  - Better science
  - Better systems (bigger, faster, more accurate, more integrated, more efficient, less wasteful)
  - Better management of global resources (food, water, energy, people time, personalised health care)
  - Healthier, longer, more enriching lives (Morris, 'News from nowhere")

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law




[indico.cern.ch]

# Diolch yn fawr!

# Backup

# Regression

Linear regression: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$

- $\epsilon \sim N(0, \sigma^2 \boldsymbol{I})$, prior: $\boldsymbol{\beta} \sim N(0, \boldsymbol{\Sigma})$, $\boldsymbol{X}$ is $n \times p$
- Negative log posterior: $(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^2/\sigma^2 + \boldsymbol{\beta}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}$
- $\mathbb{E}[\boldsymbol{\beta}|\boldsymbol{y}] = (\boldsymbol{X}'\boldsymbol{X}/\sigma^2 + \boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{X}'\boldsymbol{y}/\sigma^2$
- Prediction: $\mathbb{E}[\boldsymbol{X}_*\boldsymbol{\beta}|\boldsymbol{y}] = \boldsymbol{X}_*(\boldsymbol{X}'\boldsymbol{X}/\sigma^2 + \boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{X}'\boldsymbol{y}/\sigma^2 = \boldsymbol{S}\boldsymbol{y}$
- Regression is a **smoother**, invert $p \times p$ matrix $\boldsymbol{X}'\boldsymbol{X}$

- But: $(\boldsymbol{X}'\boldsymbol{X}/\sigma^2 + \boldsymbol{\Sigma}^{-1})\boldsymbol{\Sigma}\boldsymbol{X}' = \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{X}' + \sigma^2\boldsymbol{I})/\sigma^2$
- Prediction: $\mathbb{E}[\boldsymbol{X}_*\boldsymbol{\beta}|\boldsymbol{y}] = \boldsymbol{X}_*\boldsymbol{\Sigma}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{X}' + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$
- Invert $n \times n$ matrix $\boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{X}'$. $n$ can be big, $> 10^5$

Kernel regression: $\boldsymbol{y} = \Phi(\boldsymbol{X})'\boldsymbol{\beta} + \boldsymbol{\epsilon}$

- Inner product $\boldsymbol{X}_*\boldsymbol{\Sigma}\boldsymbol{X}'$ generalised to $\Phi(\boldsymbol{X}_*)'\boldsymbol{\Sigma}\Phi(\boldsymbol{X})$, where $\Phi(\boldsymbol{X})$ is a **kernel function**

# Kernel

- **Kernel trick**: Never write kernel functions explicitly. Only need inner-product expression $k(\boldsymbol{X}_*, \boldsymbol{X}) = \Phi(\boldsymbol{X}_*)'\boldsymbol{\Sigma}\Phi(\boldsymbol{X})$
- Arbitrary (smooth) relationships can be estimated using kernel regression

- Product form: $k(\boldsymbol{X}_*, \boldsymbol{X}) = \prod_{j=1}^{p} k_j(\boldsymbol{X}_{*j}, \boldsymbol{X}_j)$ for covariate $j$
- Squared exponential: $k_j(x_{*hj}, x_{ij}) = \exp(-(x_{*hj} - x_{ij})^2/(2\ell_j^2))$

- Correlation lengths: $\{\ell_j\}_{j=1}^{p}$ must be estimated using cross-validation. Also estimate nugget variances

- Slick leave-one-out cross-validation available (cf 'hat matrix')
- Slick Kronecker product form (fast inverse!) when $\boldsymbol{X}$ is of blocked form: $k(\boldsymbol{X}, \boldsymbol{X}) = k(\boldsymbol{X}_1, \boldsymbol{X}_1) \otimes k(\boldsymbol{X}_2, \boldsymbol{X}_2)$