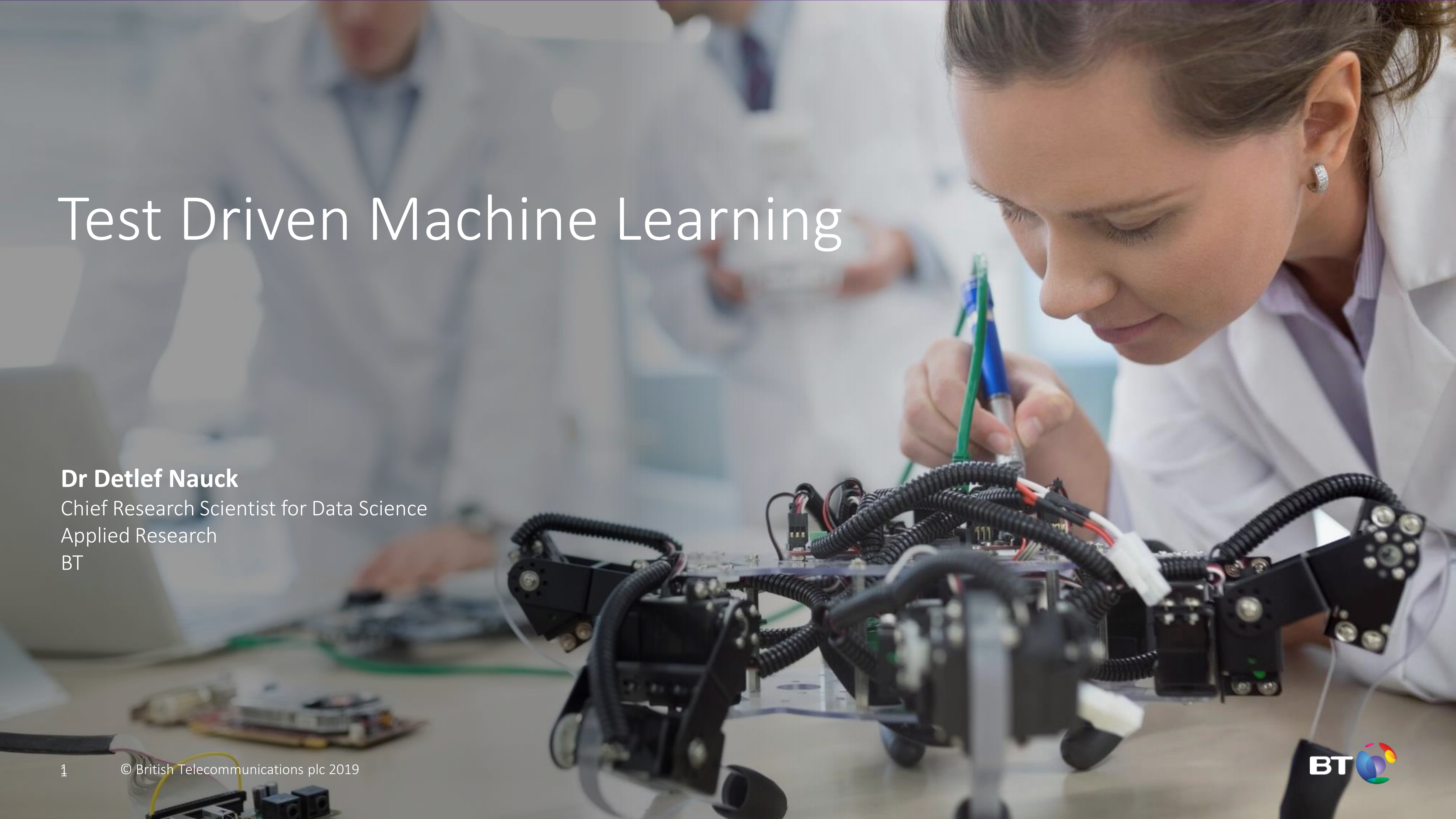
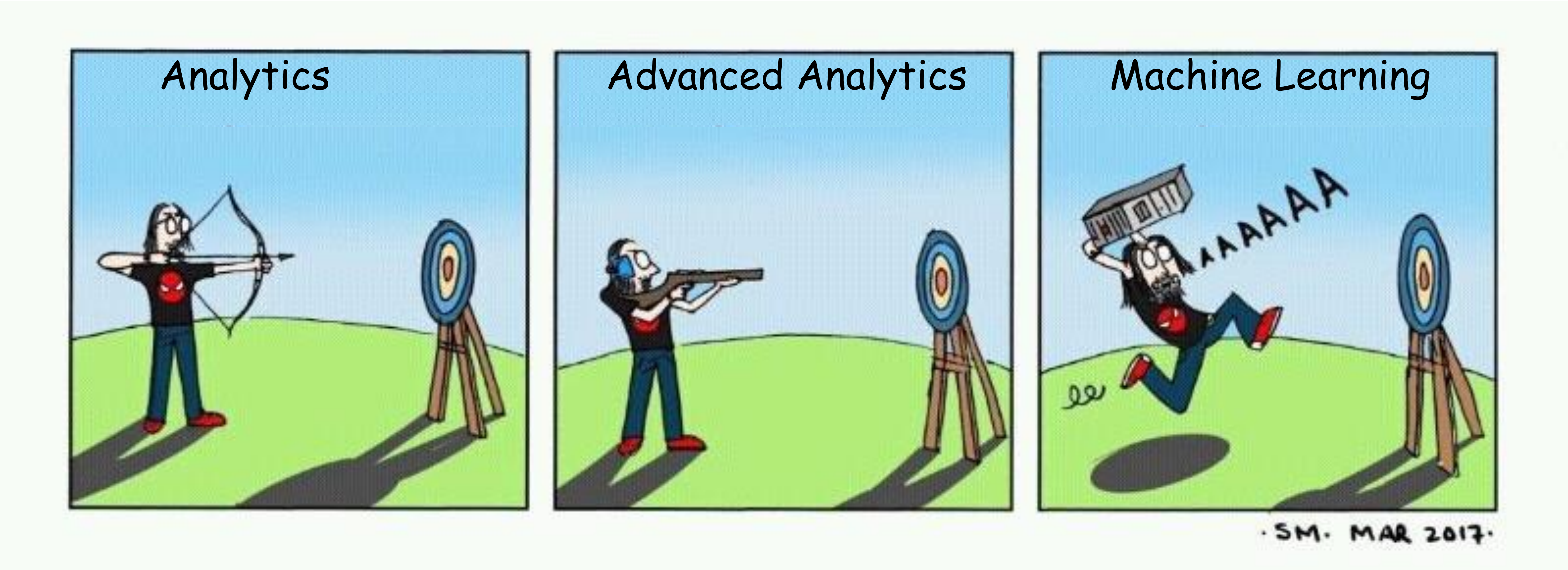


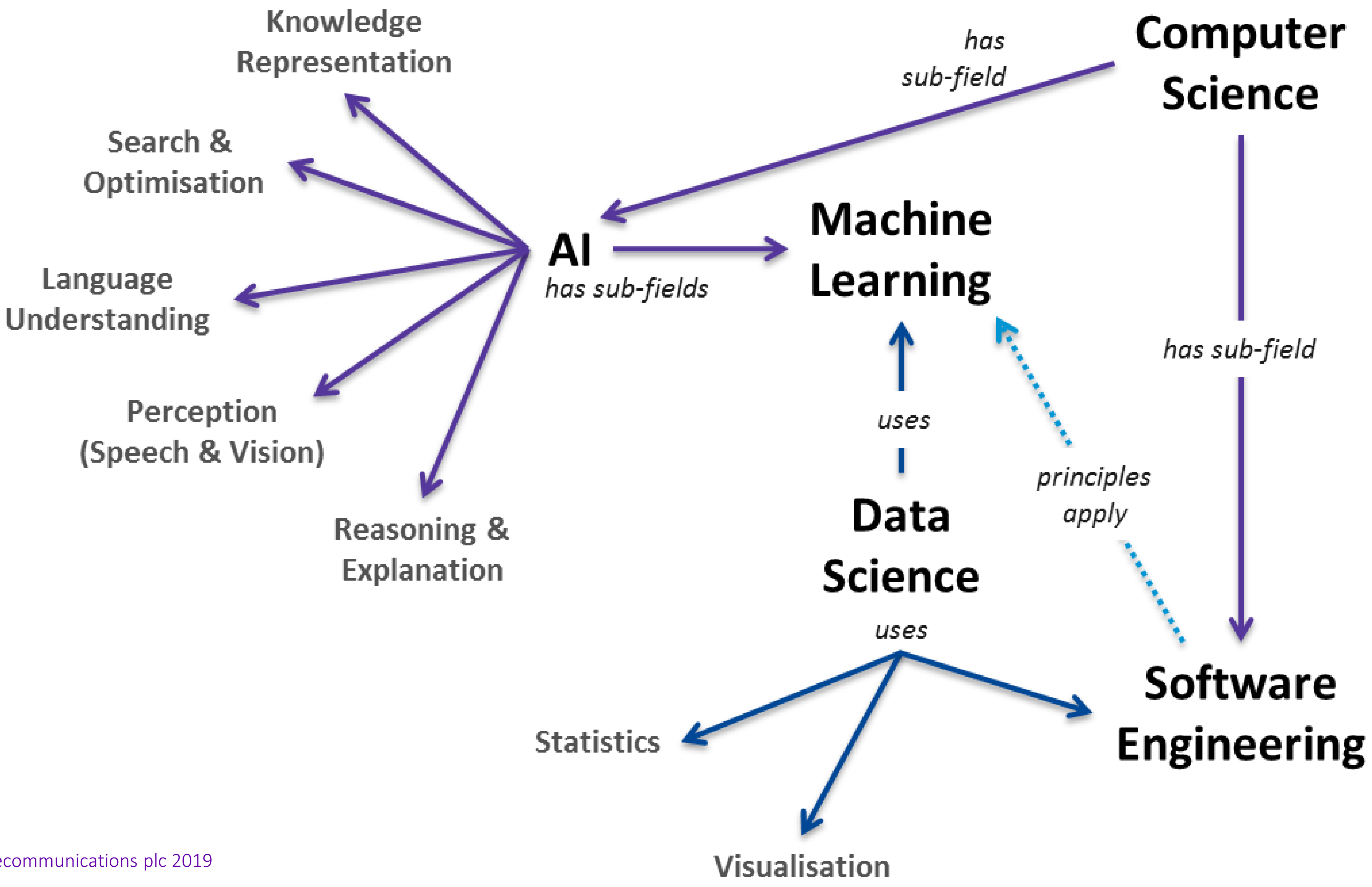
Test Driven Machine Learning

Dr Detlef Nauck
Chief Research Scientist for Data Science
Applied Research
BT



https://www.instagram.com/ground.control.toons

AI – ML – DS Taxonomy



So what are AI, Machine Learning and Data Science?

AI

“Artificial intelligence is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment.”

Nils J. Nilsson, *The Quest for Artificial Intelligence: A History of Ideas and Achievements*
(Cambridge, UK: Cambridge University Press, 2010).

Machine Learning

Field in Computer Science and AI that looks for algorithms that can automatically improve their performance at a task without explicit programming but by observing relevant data.

Data Science

Data Science uses statistics and *machine learning* for (data) analytics – the *scientific process of turning data into insight for making better decisions.*

(INFORMS - The Institute for Operations Research and the Management Sciences)

The Ethical Dimension of AI, ML and Data Science

What should AI and Data Science be used / not used for and what is the impact on society such as privacy and automation of jobs?

GDPR

For example no black box models if decisions significantly affect a person. Organisations must be able to explain their algorithmic decision making in certain conditions.

Bias – (unfair) under / over-representation of subgroups in your data

Bias in data and models not only makes models perform worse, it can also damage a brand.

http://www.wsj.com/articles/wisconsin-supreme-court-to-rule-on-predictive-algorithms-used-in-sentencing-1465119008

https://www.wired.com/story/how-coders-are-fighting-bias-in-facial-recognition-software/

▲ We might be tempted to call them 'frankenalgos' - though Mary Shelley couldn't have made this up. Illustration: Marco Goran Romano

Franken-algorithms: the deadly consequences of unpredictable code

The death of a woman hit by a self-driving car highlights an unfolding technological crisis, as code piled on code creates 'a universe no one fully understands'

https://www.theguardian.com/technology/2018/aug/29/coding-algorithms-frankenalgos-program-danger

A Report from the Coalface: Naïve Bayes – From R to Spark ML

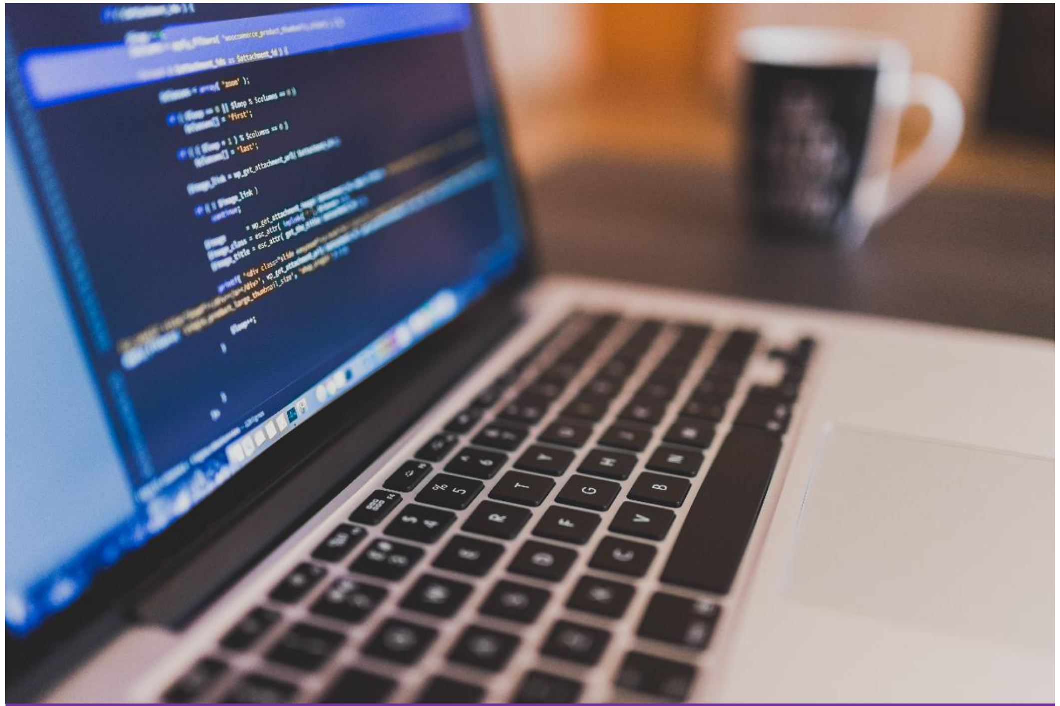
Teams builds a naïve Bayes classifier in R (e1071 package) using numerical and categorical features

Team had to rebuild the model in Spark ML which can only use categorical features encoded as numbers.

Team interpreted this as any numbers are fine, i.e. numerical features can be used too.

The team converted the categorical features into numbers and kept the numerical features as they were

Fortunately, there were negative values in the numerical features which the Spark ML method refused to accept.



Things can get tricky when you switch data analysis methods and environments without proper testing.

A Report from the Coalface: Naïve Bayes in R

naiveBayes	*Naive Bayes Classifier*

Description

Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.

Usage

```
## S3 method for class 'formula'
naiveBayes(formula, data, laplace = 0, ..., subset, na.action = na.pass)
## Default S3 method:
naiveBayes(x, y, laplace = 0, ...)


## S3 method for class 'naiveBayes'
predict(object, newdata,
  type = c("class", "raw"), threshold = 0.001, eps = 0, ...)
```

Arguments

`x`	A numeric matrix, or a data frame of categorical and/or numeric variables.
`y`	Class vector.
`formula`	A formula of the form `class ~ x1 + x2 +` .... Interactions are not allowed.
`data`	Either a data frame of predictors (categorical and/or numeric) or a contingency table.
`laplace`	positive double controlling Laplace smoothing. The default (0) disables Laplace smoothing.

A Report from the Coalface: Naïve Bayes in Spark ML

class **pyspark.mllib.classification.NaiveBayesModel**(*labels*, *pi*, *theta*)

Model for Naive Bayes classifiers.

Parameters:
- **labels** – List of labels.
- **pi** – Log of class priors, whose dimension is C, number of labels.
- **theta** – Log of class conditional probabilities, whose dimension is C-by-D, where D is number of features.

```
>>> data = [
...     LabeledPoint(0.0, [0.0, 0.0]),
...     LabeledPoint(0.0, [0.0, 1.0]),
...     LabeledPoint(1.0, [1.0, 0.0]),
... ]
>>> model = NaiveBayes.train(sc.parallelize(data))
>>> model.predict(array([0.0, 1.0]))
0.0
>>> model.predict(array([1.0, 0.0]))
1.0
```

Some Learning Algorithms Complain about Wrong Data

MLR is very easy to use. First you create a learning task where you specify the data you want to use and the target column within that data set.

```
In [14]: task = makeClassifTask(data = fw_agg_f, target = "delayed")
```

Then you create a learner by choosing a learning algorithm (here we pick a logistic regression) and specify additional parameters. Here we require the logistic regression to produce a probability for each output instead of simply producing 0 and 1. We need this later for the ROC analysis and testing a variable decision threshold.

```
In [15]: lda_lrn = makeLearner("classif.lda", id = "lda", predict.type = "prob")
```

Now we use a conveniece function in MLR to create a model by running a 10-fold cross validation and computing statistics for a number of performance measures (mmce = mean model classification error, tp = true positives, fp = false positives, tn = true negatives, fn = false negatives, auc = area under the (ROC) curve). The function will print out the performance for each fold and the average performance which we will take as an estimate for the performance on unseen data.

```
In [16]: r = crossval(lda_lrn, task, iters = 10, measures = list(mmce, tp, fp, tn, fn, auc))

[Resample] cross-validation iter 1:

Error in lda.default(x, grouping, ...): variable 1 appears to be constant within groups
Traceback:

1. crossval(lda_lrn, task, iters = 10, measures = list(mmce, tp,
 .     fp, tn, fn, auc))
2. resample(learner, task, rdesc, measures = measures, models = models,
 .     keep.pred = keep.pred, show.info = show.info)
3. parallelMap(doResampleIteration, seq_len(rin$desc$iters), level = "mlr.resample",
 .     more.args = more.args)
4. mapply(fun2, ..., MoreArgs = more.args, SIMPLIFY = FALSE, USE.NAMES = FALSE)
5. (function (learner, task, rin, i, measures, weights, model, extract,
 .     show.info)
 . {
 .     setSlaveOptions()
 .     if (show.info)
 .         messagef("[Resample] %s iter %i: ", rin$desc$id, i, .newline = FALSE)
 .     train.i = rin$train.inds[[i]]
 .     test.i = rin$test.inds[[i]]
```

Machine Learning can work really well – but it doesn't always

Image recognition with deep networks is statistically impressive, but individually unreliable

(Produced by a Deep Network trained for image recognition and automatic image labelling)
Source: Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan (Google): Show and Tell: A Neural Image Caption Generator (https://arxiv.org/pdf/1411.4555.pdf)

https://xkcd.com/1838/

Challenges in Analytics (especially Machine Learning)

Data Provenance

Where does the data come from, how was it collected and can I trust it?

Data Quality

Is the data error free?

Data Bias

Does the data accurately reflect the population/situation I want to model? What is missing?

Model Bias

Driven by unconscious bias or data bias – does my model treat everybody fairly?

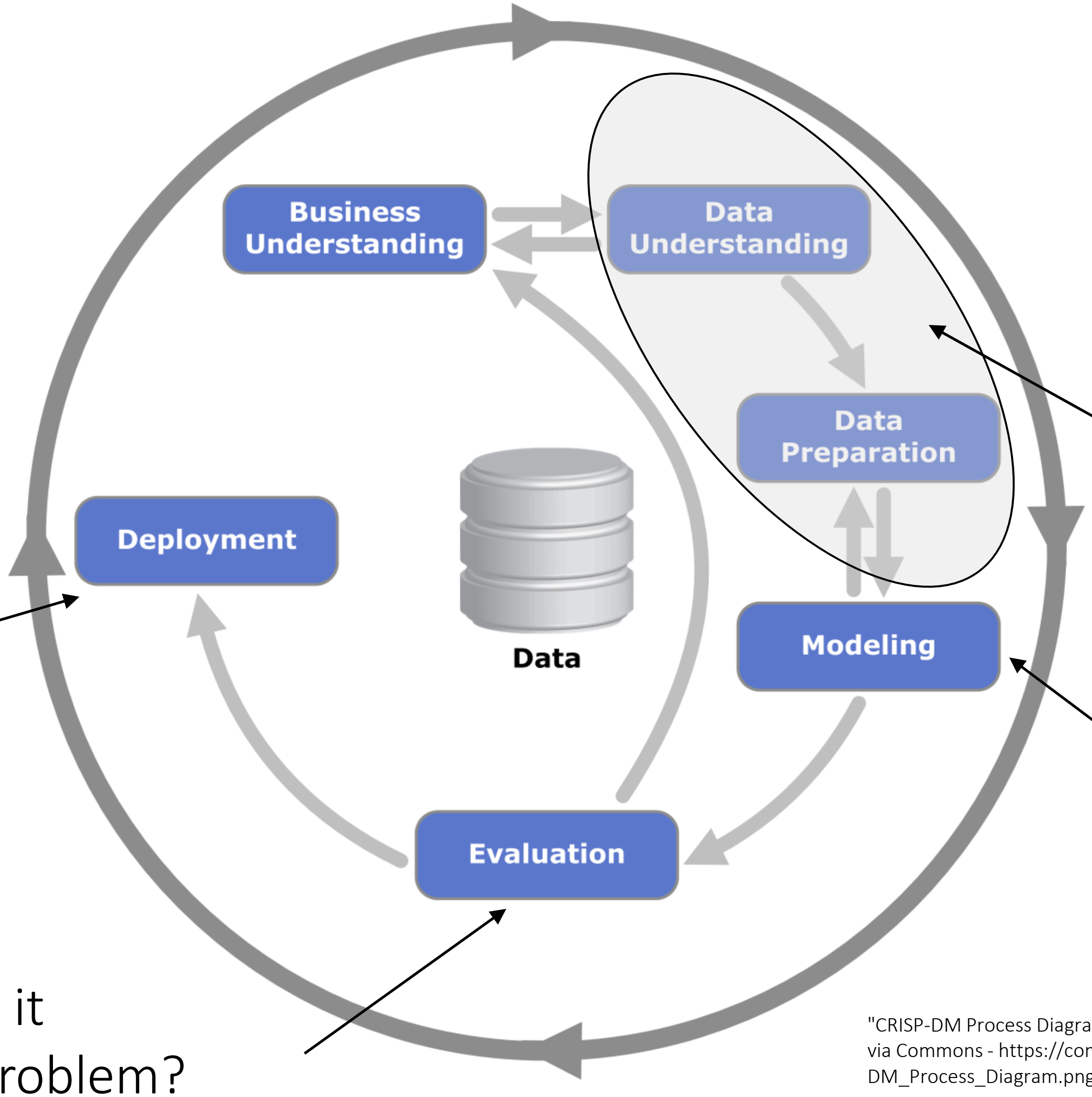
Model Comprehension

Why are the outputs of my (black box) models the way they are?

Ethics

Can I use this data? Do decisions made by models affect people? Is there discriminatory bias? …

The Process of Building Models from Data



Cross Industry Standard Process for Data Mining

Data Wrangling:
includes data access, exploratory data analysis and data transformation.
70%-90% of project effort

Analytics, **Machine Learning**: Building and testing models

95% of failed projects fail at deployment

Reality check – does it solve the business problem?

"CRISP-DM Process Diagram" by Kenneth Jensen - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png#/media/File:CRISP-DM_Process_Diagram.png

Perception Problem about Machine Learning

The AI & Machine Learning Office

This is where you would like to work

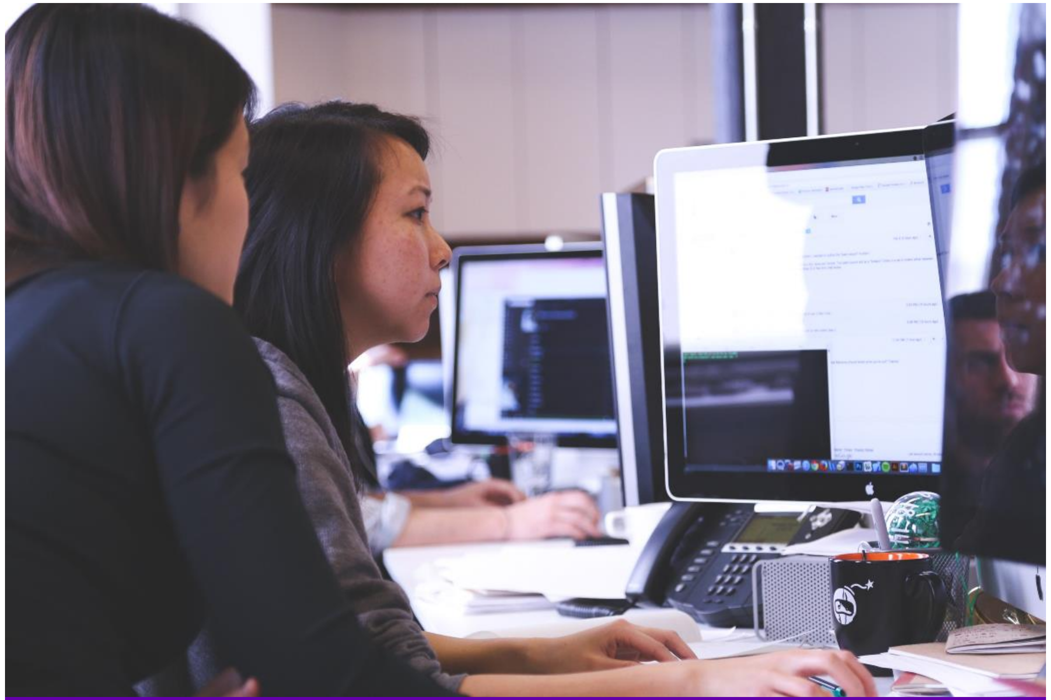
The Data Mine

This is where you will spent most of your time

Test Driven Analytics (TDA) – Learn from Software Development

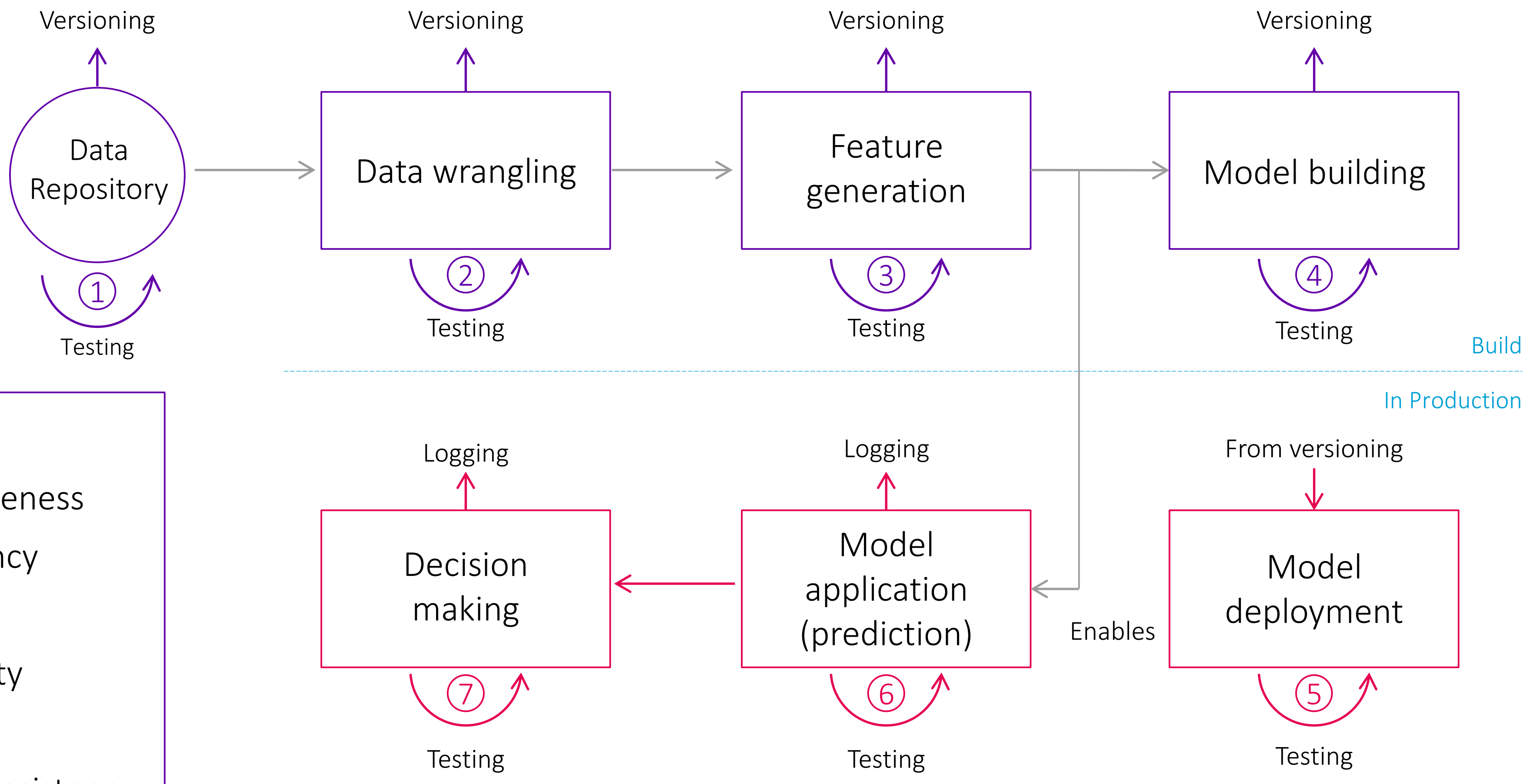
How do you make sure that your analysis is correct?

- Is your data any good?
- Have you validated your model?
- Do decisions taken have the expected business impact?



Test-driven development (TDD) is an advanced technique of using automated unit tests to drive the design of software and force decoupling of dependencies. The result of using this practice is a comprehensive suite of unit tests that can be run at any time to provide feedback that the software is still working (Microsoft: Guidelines for test-driven development)

Follow Best Practice for AI, ML and Data Science – Test Everything!



Test Scenarios

1. Data quality and completeness
2. Data errors and consistency
3. Feature quality
4. Code errors, model quality
5. Model scalability
6. Model quality, feature consistency
7. Control groups, business impact

Testing Your Model: Cross-Validation

In a nutshell:

- Partition your data into *k* ***random*** subsets of equal size
- Use *k-1* subsets to train your model and the remaining subset to test (validate) it
- Repeat *k* times, i.e. use each subset for training once
- Then, ideally, repeat the whole procedure *n* times, i.e. do this on *n* different partitions.

In practice

- *k=10* or *k=5*, and *n=1*
- Aim for *k=10* and *n>1*

Why are we doing this?

- We want to know how strongly the selection of training data impacts model building
- We want an estimate on how the model might perform on future data

Cross-Validation: What to look for

Compare the performance statistics of training and test (validation) sets

- You are looking for the training performance to be better (but only slightly) than the test performance
- You are looking for consistent figures across all folds
- If training performance is much better than test the model is overfitting
- If the test performance is much better than the training performance something is wrong (e.g. data leakage)
- If the figures vary a lot you probably do not have enough data or the wrong data

Check the model structure in each fold

- You want the structure (selected features, discovered rules, …) to ideally be the same across all folds
- Different model structures mean that data selection is influencing the model build

The Most Misused Term in Data Science, AI & Machine Learning

~~Accuracy~~

Avoid using it!

Performance of Binary Classifiers

		Confusion Matrix		
		Model		
		Yes (1)	No (0)	∑
Data	Yes (1)	True Positive (TP)	False Negative (FN – Type II error)	Condition Positive (CP)
	No (0)	False Positive (FP – Type I error)	True Negative (TN)	Condition Negative (CN)
	∑	Predicted Condition Positive (PCP)	Predictive Condition Negative (PCN)	Total Sample (N)

All Data:	N = (TP + FN + FP + TN) = (CP + CN) = (PCP + PCN)
Base Rate (Prevalence):	CP/N
Accuracy:	(TP + TN) / N
Error:	(FP + FN) / N
True Positive Rate (Recall, Detection Rate, Sensitivity):	TP / (TP + FN)
False Positive Rate (Fall Out):	FP / (FP +TN)
True Negative Rate (Specificity):	TN/(FP+TN) = 1 – FPR
Positive Predictive Value (Precision):	TP / (TP + FP)

Test Your Data!

While you pre-process your data or while you conduct exploratory data analysis you should gain some insight into properties of the data.

Think about which properties are essential and which you may want to check or guarantee before you conduct any further analysis.

Write some code to demonstrate that your data is correct.

assertr – verify() tests conditions on a data frame

```
mydata %>%
  verify(delayed == 1 | delayed == 0) %>%
  verify (sum(delayed)/nrow(mydata) >= 0.5)%>%
  summarise(count.delayed = sum(delayed), delayed_ratio = sum(delayed)/n())
```

count.delayed <int>	delayed_ratio <dbl>
2507	0.8609203

1 row

… but if it fails it cannot point to the offending value

```
mydata %>%
  verify(delayed == 1 | delayed == 0) %>%
  verify (sum(delayed)/nrow(mydata) >= 0.5)%>%
  summarise(count.delayed = sum(delayed), delayed_ratio = sum(delayed)/n())
```

```
verification [sum(delayed)/nrow(mydata) >= 0.5] failed! (1 failure)

    verb redux_fn                       predicate column index value
1 verify       NA sum(delayed)/nrow(mydata) >= 0.5     NA     1    NA
```

```
Error: assertr stopped execution
```

assertr – assert() tests a predicate on a list of columns

```
mydata %>%
  assert(within_bounds(0,Inf), everything()) %>%
  summarise(count.delayed = sum(delayed))
```

```
Column 'dewp' violates assertion 'within_bounds(0, Inf)' 36 times
    verb redux_fn              predicate column index value
1 assert       NA within_bounds(0, Inf)   dewp   410 -0.04
2 assert       NA within_bounds(0, Inf)   dewp   411 -2.92
3 assert       NA within_bounds(0, Inf)   dewp   412 -5.98
4 assert       NA within_bounds(0, Inf)   dewp   413 -0.94
5 assert       NA within_bounds(0, Inf)   dewp   414 -4.00
  [omitted 31 rows]
```

```
Error: assertr stopped execution
```

The function is internally using the select() function from dplyr

assertr – insist() dynamically determines a predicate function

```
mydata %>%
  insist(within_n_sds(3), wind_speed, wind_gust) %>%
  summarise(count.delayed = sum(delayed))
```

```
Column 'wind_speed' violates assertion 'within_n_sds(3)' 18 times
    verb redux_fn       predicate     column index    value
1 insist       NA within_n_sds(3) wind_speed   571 35.67418
2 insist       NA within_n_sds(3) wind_speed   573 35.67418
3 insist       NA within_n_sds(3) wind_speed   574 42.57886
4 insist       NA within_n_sds(3) wind_speed   577 36.82496
5 insist       NA within_n_sds(3) wind_speed   578 39.12652
  [omitted 13 rows]


Column 'wind_gust' violates assertion 'within_n_sds(3)' 18 times
    verb redux_fn       predicate    column index    value
1 insist       NA within_n_sds(3) wind_gust   571 41.05313
2 insist       NA within_n_sds(3) wind_gust   573 41.05313
3 insist       NA within_n_sds(3) wind_gust   574 48.99890
4 insist       NA within_n_sds(3) wind_gust   577 42.37743
5 insist       NA within_n_sds(3) wind_gust   578 45.02602
  [omitted 13 rows]
```

```
Error: assertr stopped execution
```

R Packages for Assertive Programming

Tony Fischetti's assertr
(https://github.com/tonyfischetti/assertr)

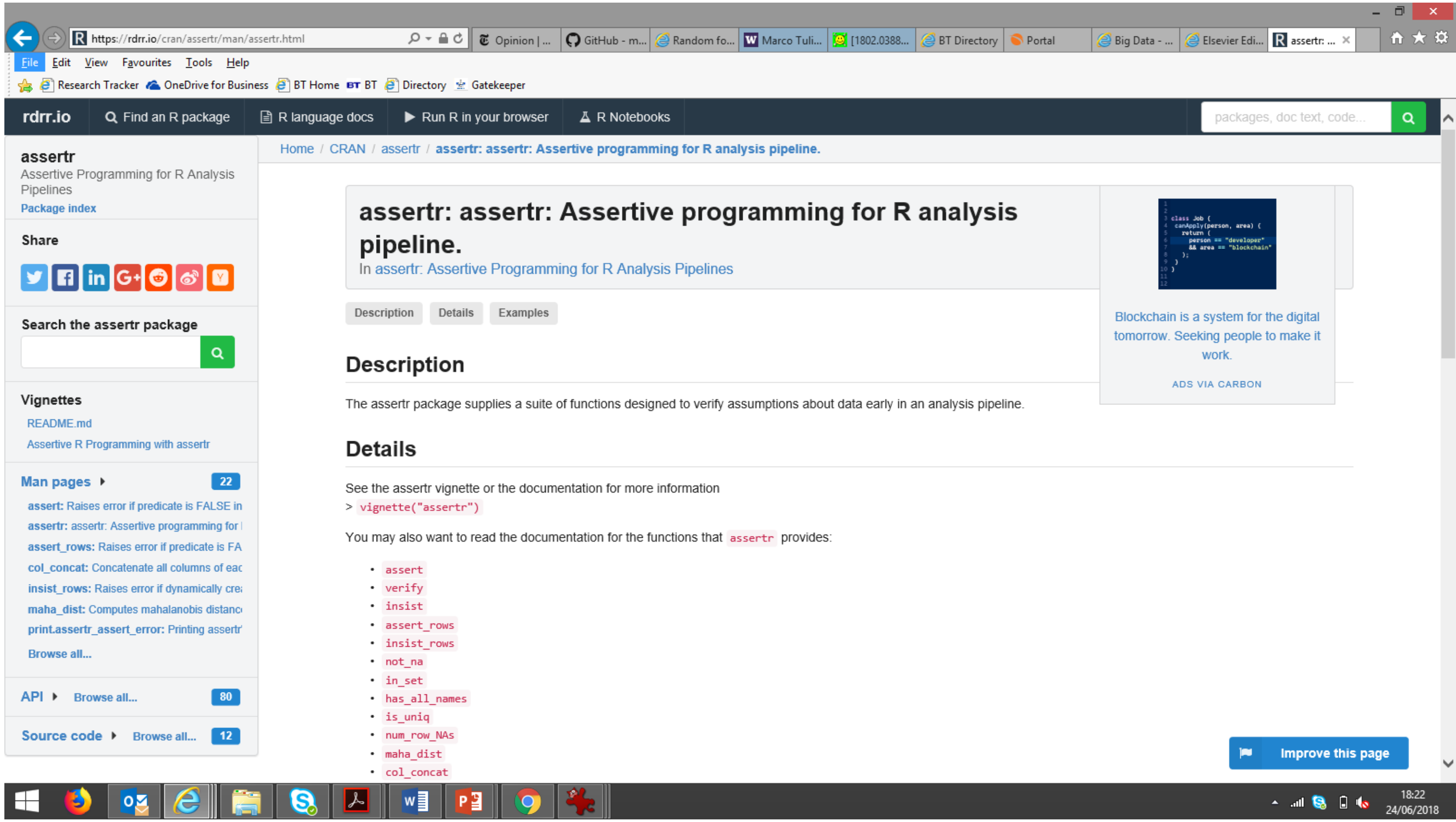
Hadley Wickham's assertthat
(https://github.com/hadley/assertthat)

Richard Cotton’s assertive
(https://bitbucket.org/richierocks/assertive)

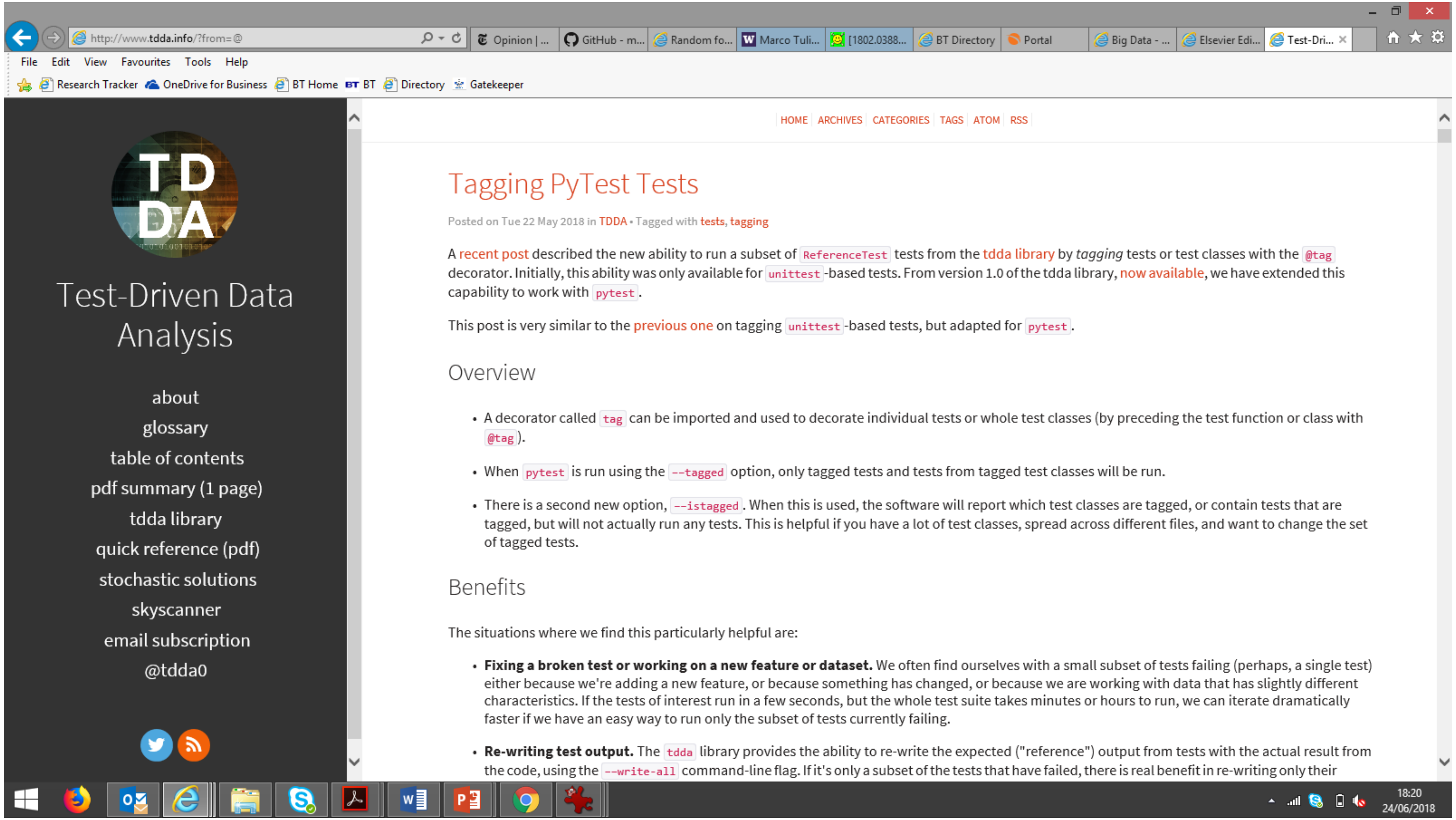
Michel Lang's checkmate
(https://github.com/mllg/checkmate)

Stefan Bache's ensurer
(https://github.com/smbache/ensurer)

Gaston Sanchez's tester
(https://github.com/gastonstat/tester)



Python – TDDA by Nick Radcliffe (http://tdda.info)



http://www.tdda.info/?from=@

HOME | ARCHIVES | CATEGORIES | TAGS | ATOM | RSS

Test-Driven Data Analysis

about
glossary
table of contents
pdf summary (1 page)
tdda library
quick reference (pdf)
stochastic solutions
skyscanner
email subscription
@tdda0

Tagging PyTest Tests

Posted on Tue 22 May 2018 in TDDA • Tagged with tests, tagging

A recent post described the new ability to run a subset of `ReferenceTest` tests from the tdda library by *tagging* tests or test classes with the `@tag` decorator. Initially, this ability was only available for `unittest`-based tests. From version 1.0 of the tdda library, now available, we have extended this capability to work with `pytest`.

This post is very similar to the previous one on tagging `unittest`-based tests, but adapted for `pytest`.

Overview

- A decorator called `tag` can be imported and used to decorate individual tests or whole test classes (by preceding the test function or class with `@tag`).
- When `pytest` is run using the `--tagged` option, only tagged tests and tests from tagged test classes will be run.
- There is a second new option, `--istagged`. When this is used, the software will report which test classes are tagged, or contain tests that are tagged, but will not actually run any tests. This is helpful if you have a lot of test classes, spread across different files, and want to change the set of tagged tests.

Benefits

The situations where we find this particularly helpful are:

- **Fixing a broken test or working on a new feature or dataset.** We often find ourselves with a small subset of tests failing (perhaps, a single test) either because we're adding a new feature, or because something has changed, or because we are working with data that has slightly different characteristics. If the tests of interest run in a few seconds, but the whole test suite takes minutes or hours to run, we can iterate dramatically faster if we have an easy way to run only the subset of tests currently failing.
- **Re-writing test output.** The `tdda` library provides the ability to re-write the expected ("reference") output from tests with the actual result from the code, using the `--write-all` command-line flag. If it's only a subset of the tests that have failed, there is real benefit in re-writing only their

Research Trends in Data Science: Supporting Tools

Data Quality

Quilt
Data Registry: data versioning and checks
http://quiltdata.com

TopNotch
quality controlling large scale data sets
https://github.com/blackrock/TopNotch

Feature Crafting

FeatureHub
register features and discover features created by others
https://github.com/HDI-Project/FeatureHub

Model Management

ModelDB
manage ML Models
https://mitdbg.github.io/modeldb/

MLflow
manage ML lifecycle
https://mlflow.org/

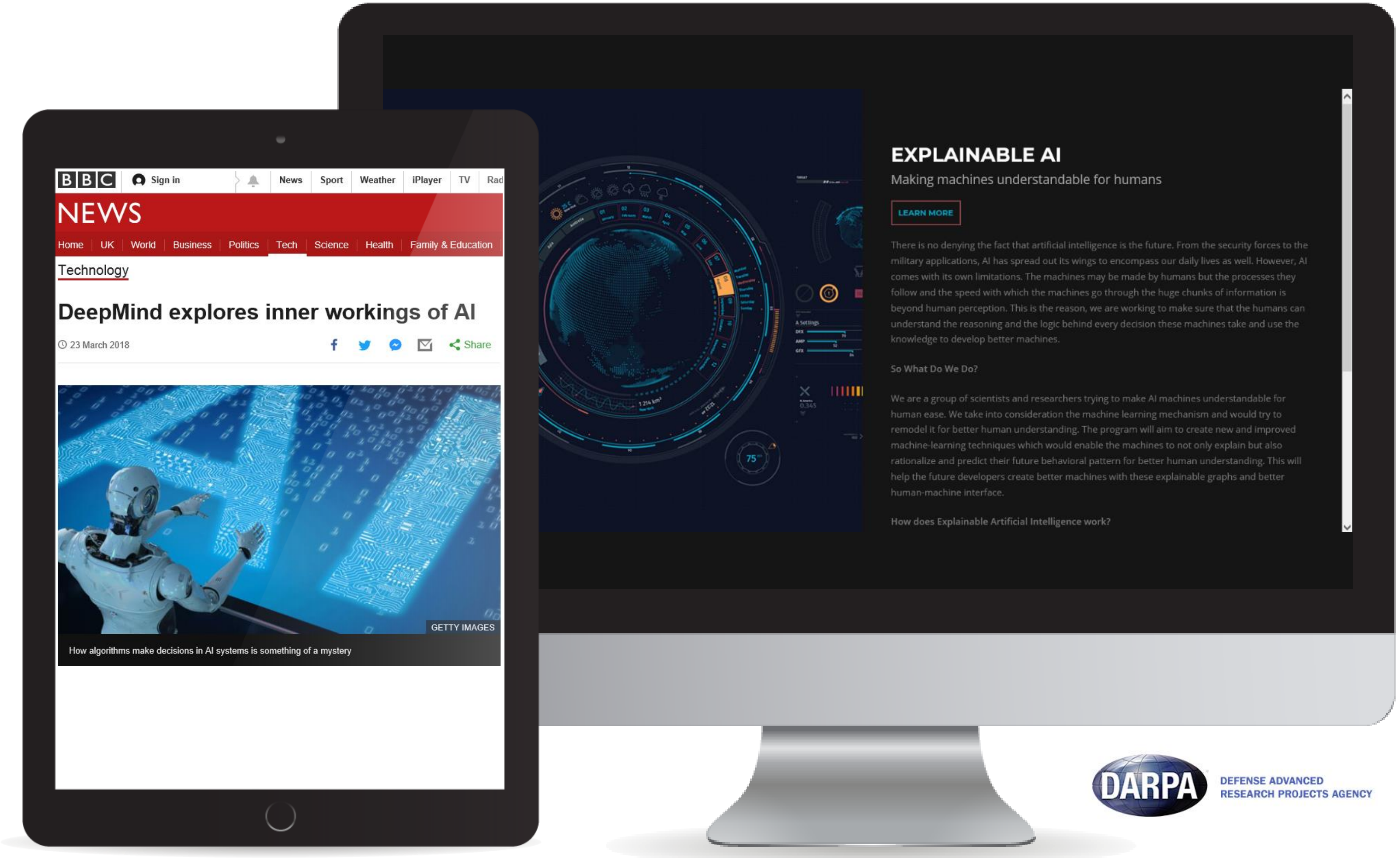
Future of AI: Explainable AI (Xai)

AI systems that will have to explain to us what they’re doing and why.

Think explainability by design.

Model-based design could be a way.

(see also http://www.darpa.mil/program/explainable-artificial-intelligence)



DARPA DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

http://www.bbc.co.uk/news/technology-43514566   http://explainableai.com

BT

BT