Applying Data Science in Commercial Environments

Dr Stephan Bloehdorn
Associate Partner, Data Science Leader DACH
Cognitive Process Transformation | Artificial Intelligence
IBM Global Business Services

@bloehdorn

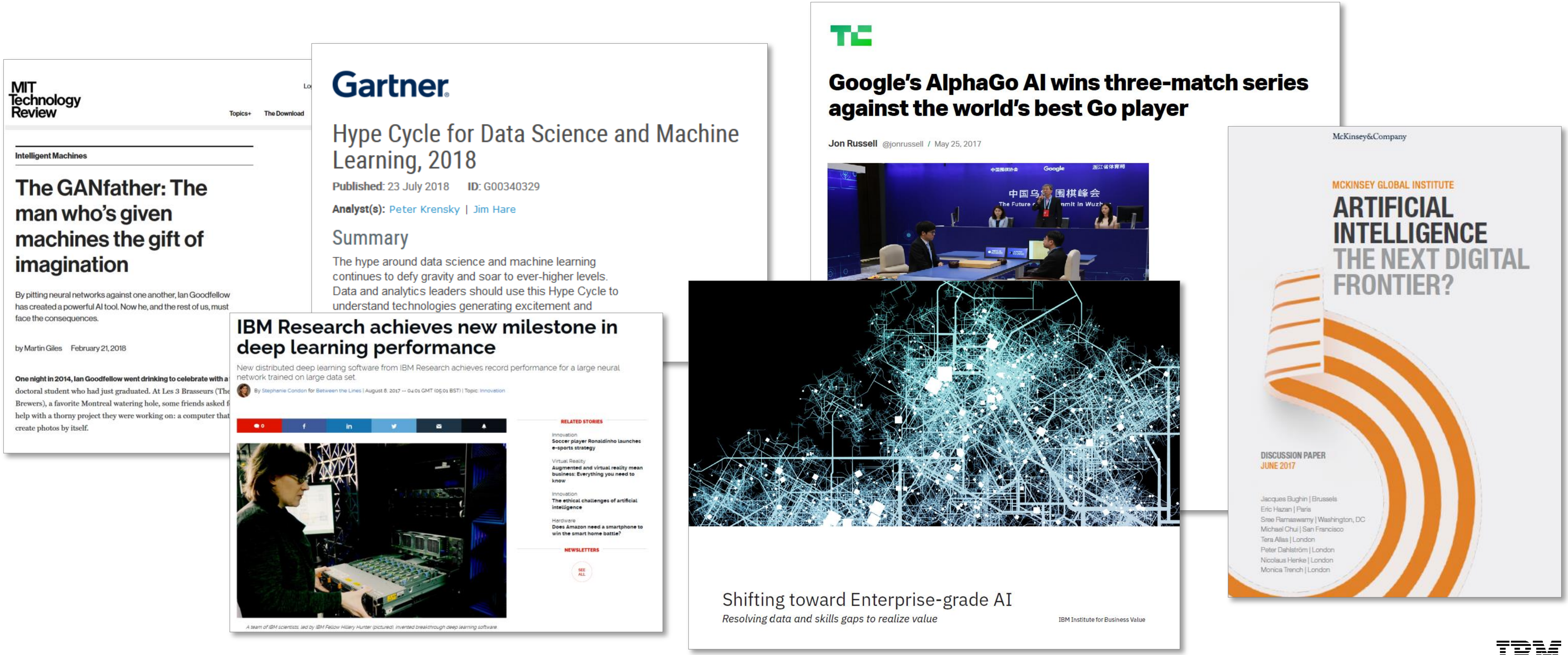
10.10.2018

We 💙 Data Science

IBM **Services**

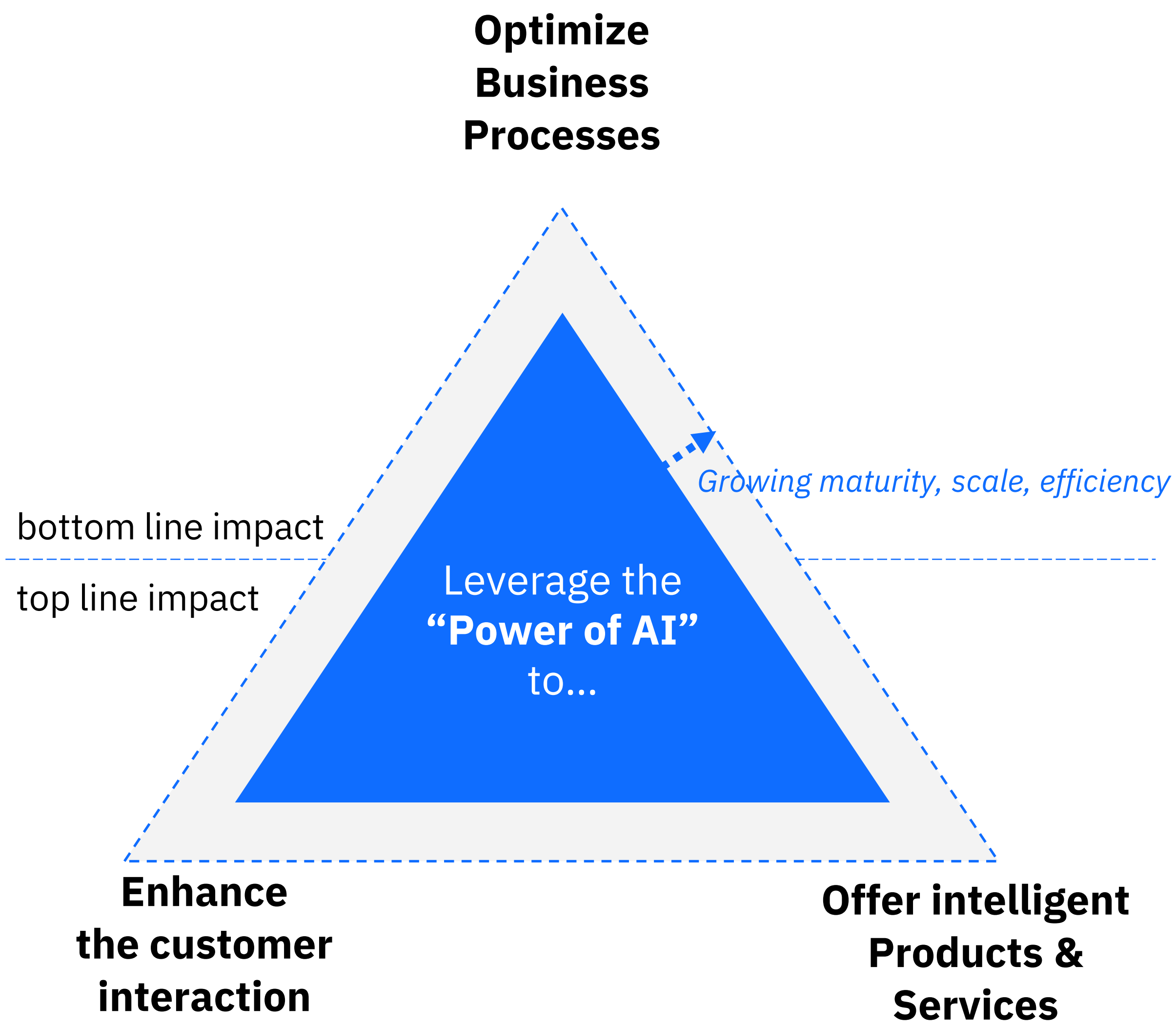
AI, Machine Learning, Data Science…
—
Buzz or Business?

Artificial Intelligence, Machine Learning, Data Science



Our mission

Artificial Intelligence is the core of the "Cognitive Enterprise"



Source: https://www.ibm.com/services/events/enterprise-ai/

Example 1

Enhance the Customer Interaction

“Developing personalized mobility services”

05.09.17 | Wolfsburg/Berlin | Technologie

Volkswagen und IBM entwickeln gemeinsam digitale Mobilitätsdienste

Pressekontakte

- Vereinbarung auf fünf Jahre ausgelegt
- Einsatz von IBM Hybrid Cloud zur Unterstützung des digitalen Ecosystems Volkswagen WE
- Vernetzung der Fahrzeuge mit Fahrer und Umfeld im Fokus

Volkswagen und IBM haben heute angekündigt, gemeinsam digitale Mobilitätsdienste zu entwickeln. Jürgen Stackmann, Vertriebsvorstand der Marke Volkswagen: „Ziel der fünfjährigen Vereinbarung zwischen Volkswagen und IBM ist es, personalisierte digitale Dienstleistungen für den Fahrer zu entwickeln und damit den Trend der zunehmenden Vernetzung zwischen Fahrzeugen und Fahrern aktiv zu gestalten."

https://www.volkswagen-newsroom.com/de/pressemitteilungen/volkswagen-und-ibm-entwickeln-gemeinsam-digitale-mobilitaetsdienste-1602

It’s really about augmented intelligence – helping us to make better, faster, more decisions

HR | Finance | Logistics | Operations | Industry Specific Processes

Example 2:
Optimize Business Processes

“Making the corporate live of employees and HR professionals less stressful”

CARL Your smart HR assistant @Siemens

Hello, how can I help?

Chat
for HR questions in natural language

Curated content & Smart Search
to leverage HR specialist knowledge

Intranet Search

Real HR agent integration
to create a seamless user experience

Continuously learning and becoming more personalized

Enabling a premium vehicle advisor functionality requires a comprehensive AI foundation

At Home

Daniel talks to Alexa about the weather & today's meetings

Daniel's car is already preset for his business trip

At Coffee Shop

Daniel picks up his latte macchiato at Mario's Café without waiting as his assistant made a pre-order

Dan, a busy father of two kids is married to Suzy.

In the City

Daniel takes a Snap like shuttle that comes with his preferred massage seats

At Work

Daniel requests his assistant to take care of a restaurant reservation

At the Restaurant

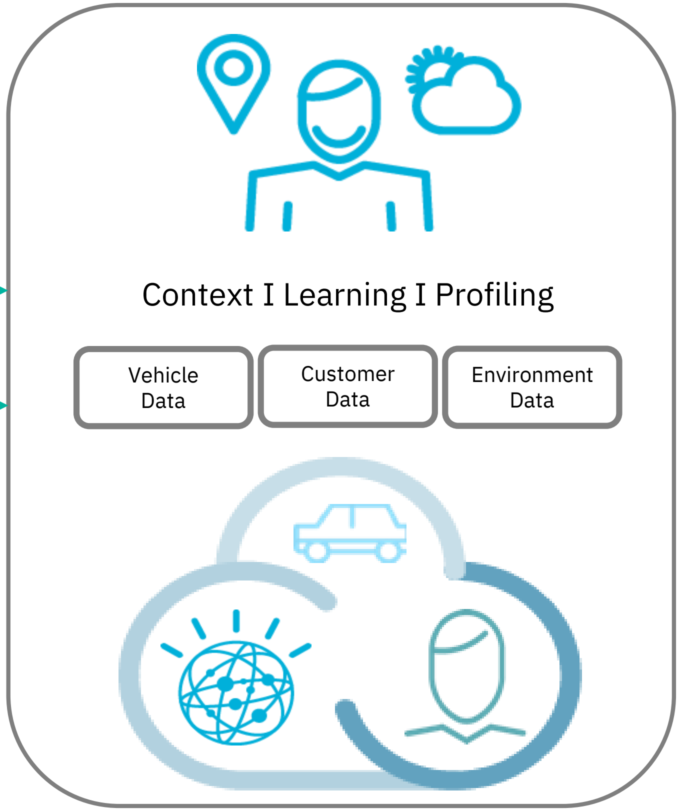
Daniel's gets a privileged service

Continuous Machine Learning is a key enabler to provide the best experience

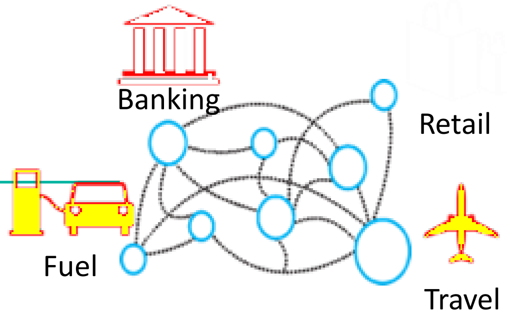
Intelligent Mobility Assistants

Premium in-car experience

Continuous Machine Learning & Recommendation Engine



Multi-Cloud Approach



Merchant Ecosystem

Customer Behaviour

Data Science Solution Engineering

—

Lab or Live?

Best Practices for Data Science Projects

Best practices for building accurate models are well understood…

Example: CRISP-DM Cross Industry Standard Process for Data Mining

… but less so for building productive Data Science solution at scale.

Holistic Architecture	Effective Engineering	Smooth Operations
Application Logic Technical Integration Model Management Tracing, Logging, Metrics	Standards Pipelines Automation	Technical Monitoring Model Monitoring Maintenance Strategy

High-Performing Team

Targeted Project Approach

* Typically this means *initial* models

Considerations for successfully engineering(complex) Machine Learning solutions in production are manifold…

Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.

Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison
{ebner,vchaudhary,mwyoung,jfcrespo,dennison}@google.com
Google, Inc.

Abstract

Machine learning offers a fantastically powerful toolkit for building useful complex prediction systems quickly. This paper argues it is dangerous to think of these quick wins as coming for free. Using the software engineering framework of *technical debt*, we find it is common to incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns.

1 Introduction

As the machine learning (ML) community continues to accumulate years of experience with live systems, a wide-spread and uncomfortable trend has emerged: developing and deploying ML systems is relatively fast and cheap, but maintaining them over time is difficult and expensive.

This dichotomy can be understood through the lens of *technical debt*, a metaphor introduced by Ward Cunningham in 1992 to help reason about the long term costs incurred by moving quickly in software engineering. As with fiscal debt, there are often sound strategic reasons to take on technical debt. Not all debt is bad, but all debt needs to be serviced. Technical debt may be paid down by refactoring code, improving unit tests, deleting dead code, reducing dependencies, tightening APIs, and improving documentation [8]. The goal is *not* to add new functionality, but to enable future improvements, reduce errors, and improve maintainability. Deferring such payments results in compounding costs. Hidden debt is dangerous because it compounds silently.

In this paper, we argue that ML systems have a special capacity for incurring technical debt, because they have all of the maintenance problems of traditional code plus an additional set of ML-specific issues. This debt may be difficult to detect because it exists at the *system* level rather than the code level. Traditional abstractions and boundaries may be subtly corrupted or invalidated by the fact that data influences ML system behavior. Typical methods for paying down code level technical debt are not sufficient to address ML-specific technical debt at the system level.

This paper does not offer novel ML algorithms, but instead seeks to increase the community's awareness of the difficult tradeoffs that must be considered in practice over the long term. We focus on system-level interactions and interfaces as an area where ML technical debt may rapidly accumulate. At a system-level, an ML model may silently erode abstraction boundaries. The tempting re-use or chaining of input signals may unintentionally couple otherwise disjoint systems. ML packages may be treated as black boxes, resulting in large masses of "glue code" or calibration layers that can lock in assumptions. Changes in the external world may influence system behavior in unintended ways. Even monitoring ML system behavior may prove difficult without careful design.

1

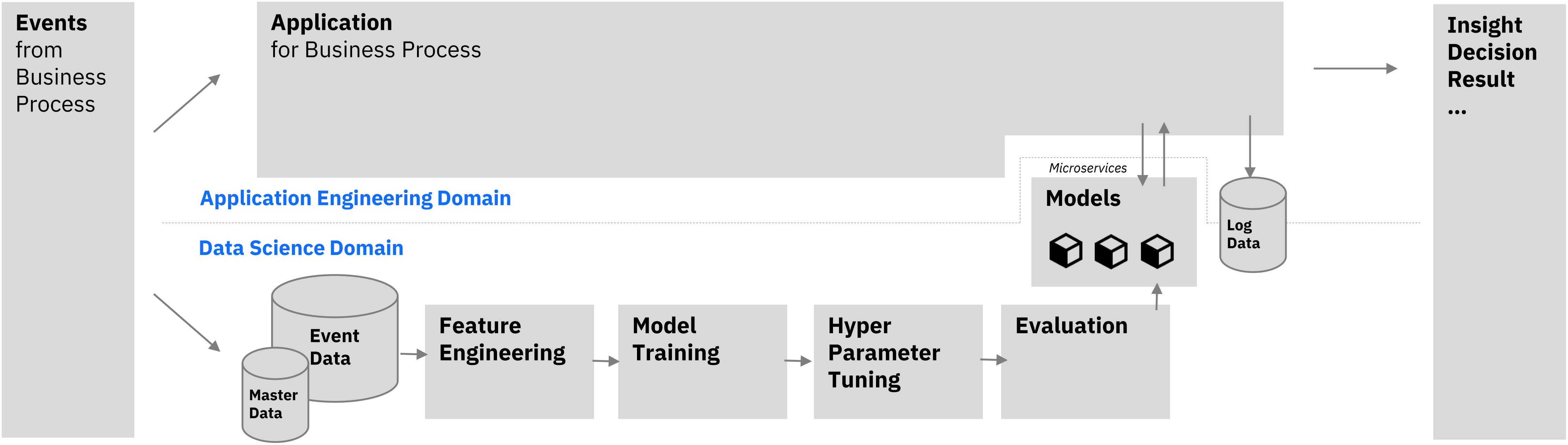
Exemplary Observations:

Complex Models Erode Boundaries
Data Dependencies Cost More than Code Dependencies
Feedback Loops
System Anti-Patterns
Configuration Debt
Dealing with Changes in the External World
…

Source: Sculley et al: Hidden Technical Debt in Machine Learning Systems NIPS'15

Data Science Solutions
—
Holistic Architecture

Architecting the logical & technical integration with the business application is integral part of the Data Science Solution



Key Considerations

- **Logical interdependency** of application – model – data
- **Technical integration** of model into application (e.g. microservices, containers)
- Model **modularization** for encapsulation and reusability
- Model **scalability** under heavy throughput
- Systematic approach to **logging, tracing, metrics**

Model management and retraining need to be architected as integral part of any Data Science Solution

Modeling & Evaluation

Model Versioning

Model Deployment

Model Monitoring

Dynamic Model Selection & Retraining

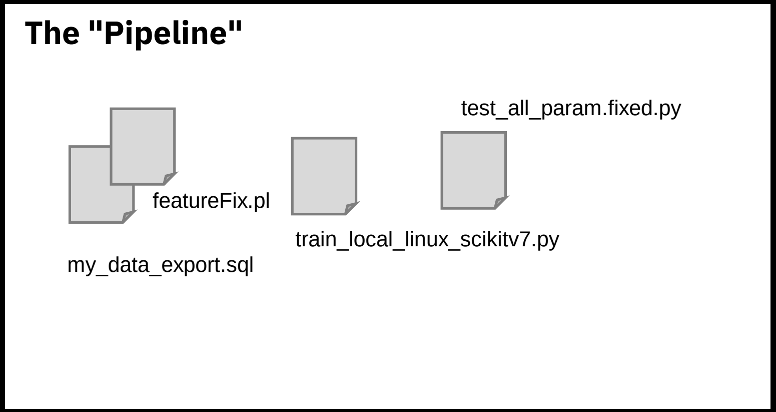
Data Science Solutions are **not** static by definition!

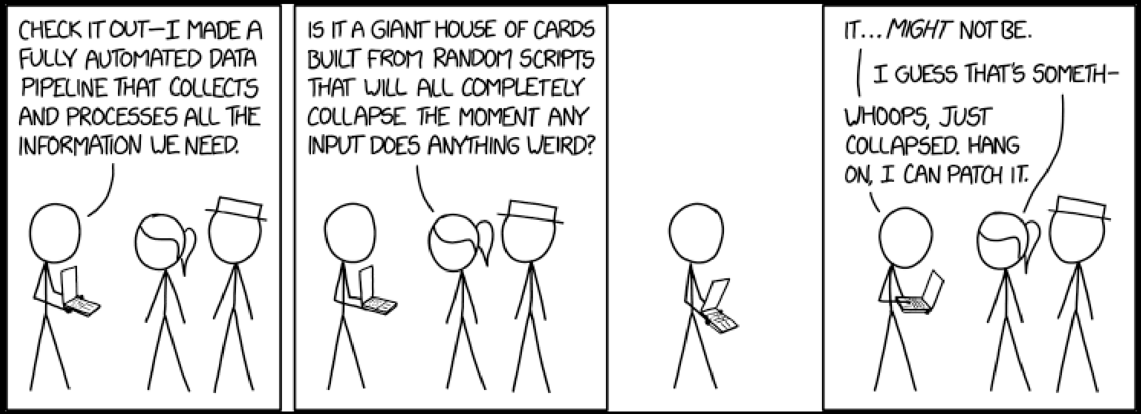
Data Science Solutions

—

Effective Engineering

Reproducibility - It ~~works~~ worked on my machine (just before, I swear!)





Robust Pipelines, Standards, Automation

Machine Learning Engineering:
Common Issues

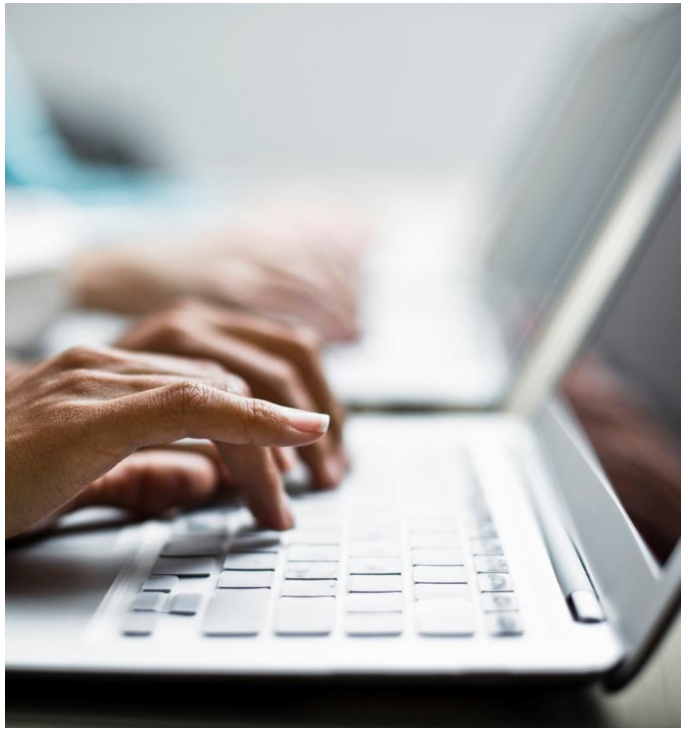
- Reproducibility issues
- Portability issues
- Scalability issues
- Debugging issues
- Reusability issues
- No automation

Machine Learning Engineering:
Best Practices

- Common processing and machine learning framework
- Separate data from code
- Modularized pipeline operations: e.g. raw data loading, feature building, training, hyperparameter tuning, evaluation…
- Naming standards for data model, machine learning model, pipeline operations
- Standardized unit tests
- Heavy automation

Testing

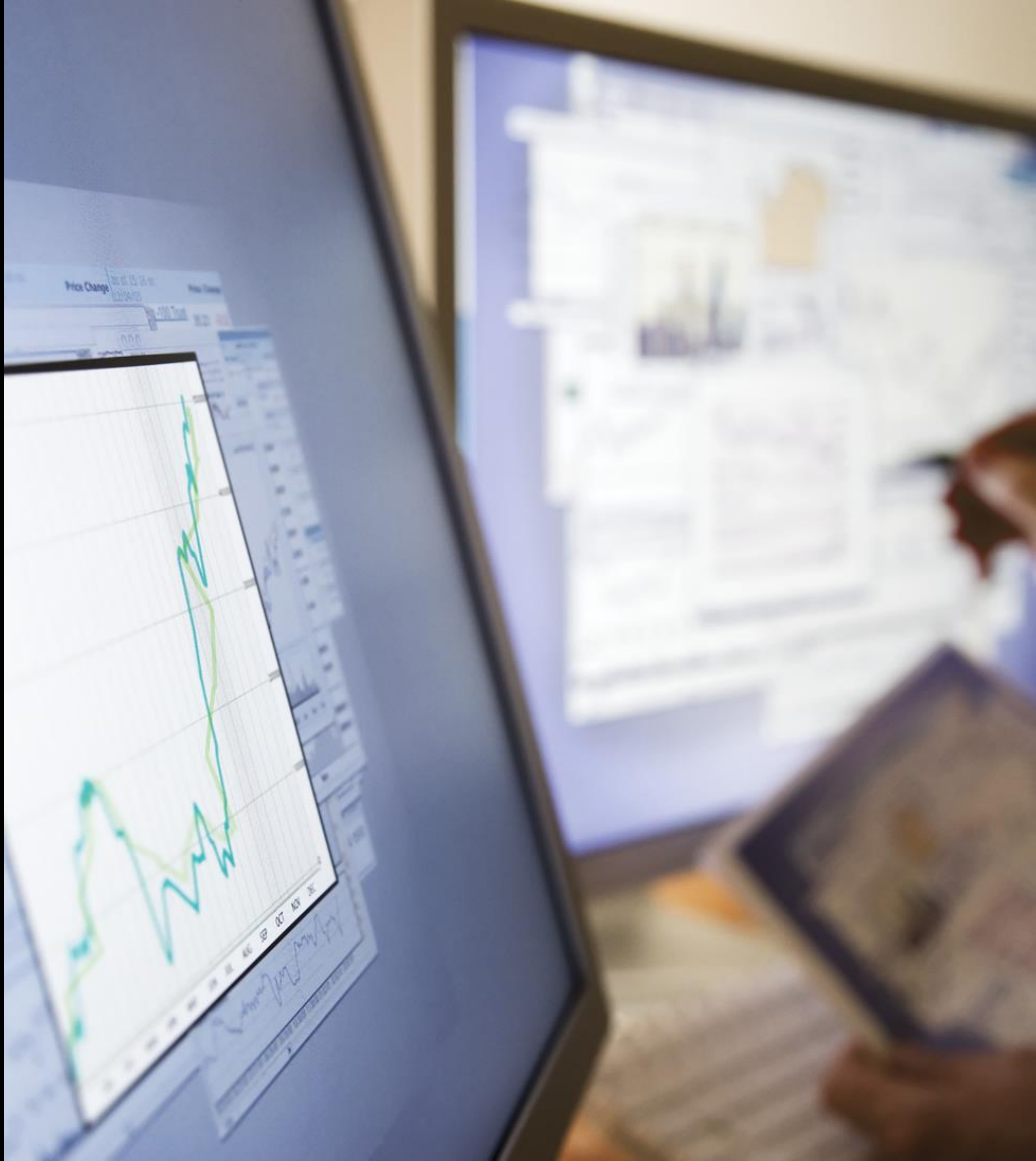
Tests for Features and Data	▪ Distributions of each feature ▪ Features are same in both the training and serving stack ▪ Relationship between different features and targets ▪ Privacy control in model training ▪ Cost of computing each feature ▪ Does not contain features determined unsuitable for use ▪ Time to add new features to production
Tests for Model Development	▪ Model code goes through code review ▪ Offline proxy metrics are measuring what will be A/B tested ▪ Hyperparameter tuning ▪ Effect of model staleness ▪ Simple models as a baseline ▪ Model performs well across different data slices ▪ Test for implicit bias in the model or data
Tests for ML Infrastructure	▪ Reproducibility of model training ▪ Integrations tests for the ML systems ▪ Quality tests before deployment of the model ▪ Ability to rollback deployed models ▪ Testing via a canary process
Monitoring Tests for ML Systems	▪ Upstream instability in features, both in training and serving ▪ Data invariants hold in training and serving inputs ▪ Model staleness ▪ Train/Test skew in features and inputs ▪ Slow leak regression in latency, throughput etc. ▪ Regression in prediction quality



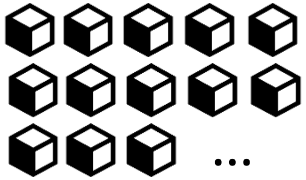
Source: https://sourabhbajaj.com/blog/2017/09/07/what-s-your-ml-test-score/#tests-for-features-and-data

Data Science Solutions
—
Smooth Operations

"Launching is easy, operation is hard"



Monitoring and Maintenance



Scalability issues?

Inconsistent model behaviour?

Concept drift?

Machine Learning Operations:
Best Practices

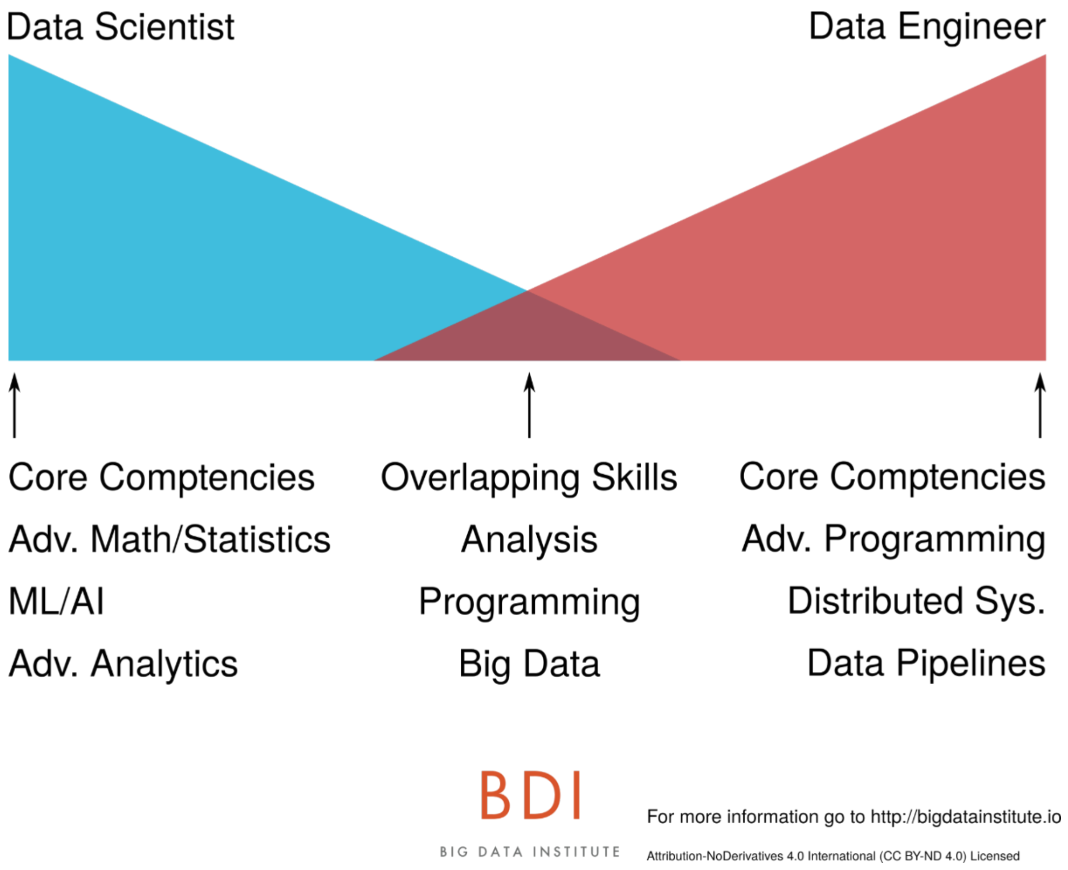
- Monitoring technical KPIs: requests, throughput, time for processing steps
- Monitoring model execution: results, confidence cores
- Monitoring outputs: class distributions vs input distributions, A/B testing, quality reviews
- Fallback strategy if model deteriorates

Data Science Solutions

—

High-Performing Team

Data Scientists vs Data Engineers / Machine Learning Engineers



Option 1 - Depending on one ML "superhero"
Anti-Pattern – not scalable, high risk

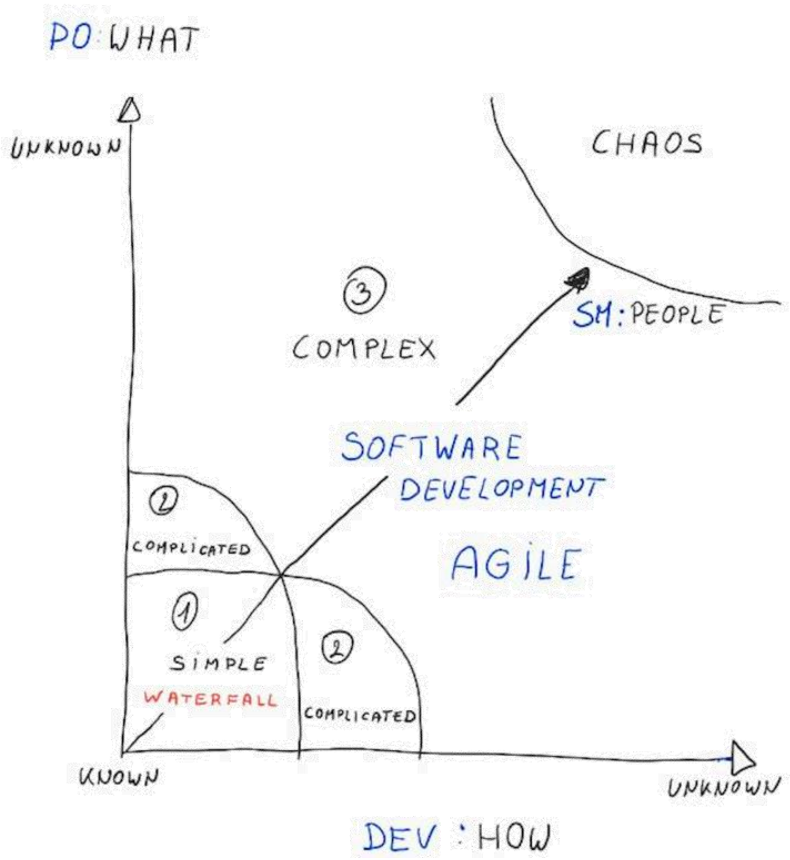
Option 3 – Agile Team
Balances scalability and quality

Option 2 - Strictly seperated roles
Researcher creates model, engineer refactors and deploys model – neither one understands output of the other
Anti-Pattern – high risk, quality issues

Sources Image http://www.bigdatainstitute.io/; Data engineers vs. data scientists; O'Reilly; https://www.oreilly.com/ideas/data-engineers-vs-data-scientists

Data Science Solutions
—
Targeted Project Approach

Agile Development for Data Science



Often we do this....

Data Science Analysis e.g. PoC 1	Improved Data Science Analysis e.g. PoC 2	Improved Data Science Analysis e.g. PoC 3	...	Great application in production

Almost always we should be doing this....

MVP 1 No model Collect data Understand usage Research	MVP 2 Simple model Collect data Understand usage Research	MVP 3 Improved model Collect data Understand usage Research	Great application in production	...

Sources Image https://medium.com/grand-parade/the-cost-of-software-development-and-tips-how-to-reduce-it-60ba44e85948

Wrapping Up
Moving On

Tesla's "Software 2.0 IDEs" notion for AD/ADAS development is very much in line with IBM's PoV for "Enterprise grade AI development"

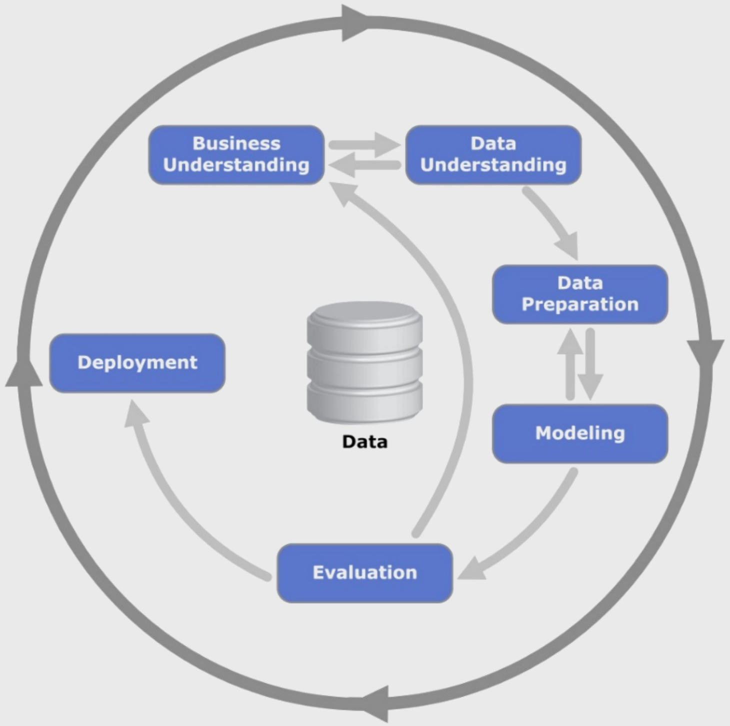
2.0 IDEs

- Show a full inventory/stats of the current dataset
- Create / edit annotation layers for any datapoint
- Flag, escalate & resolve discrepancies in multiple labels
- Flag & escalate datapoints that are likely to be mislabeled
- Display predictions on an arbitrary set of test datapoints
- Autosuggest datapoints that should be labeled
- ...

https://vimeo.com/272696002 (jump to @15:30)

Best Practices for Data Science Projects

Best practices for building accurate models are well understood...



... but less so for building productive Data Science solution at scale.

Holistic Architecture

- Application Logic
- Technical Integration
- Model Management
- Tracing, Logging, Metrics

Effective Engineering

- Standards
- Pipelines
- Automation

Smooth Operations

- Technical Monitoring
- Model Monitoring
- Maintenance Strategy

High-Performing Team

Targeted Project Approach

* Typically this means *initial* models

IBM

Resources (Selection)

IBM Data Science Community
https://community.ibm.com/community/user/datascience/home

What is hardcore data science—in practice? The anatomy of an architecture to bring data science into production.
https://www.oreilly.com/ideas/what-is-hardcore-data-science-in-practice

GCP – What is ML Ops?
https://www.youtube.com/watch?v=_jnhXzY1HCw

Google's "Rules of Machine Learning"
https://developers.google.com/machine-learning/guides/rules-of-ml/

Under the Hood of Uber's Experimentation Platform
https://eng.uber.com/xp/

IBM®