Imperial College London

Ethics of Data Science – Part III

Week 1: Reproducibility and Robustness

Deploying ML Systems vs Traditional Software

Dr. Chris Anagnostopoulos, Hon. Senior Lecturer

Reproducibility and robustness

What?

- Reproducibility in science refers to the ability to reproduce the same result in a different setting.
- In data science, it can also mean the ability to reconstruct the precise numerical results reported in the conclusions of a report or presentation from the raw data inputs used to produce them.
- An analytical pipeline is statistically *robust* when it is not overly sensitive to violations in its basic assumptions, and is robust from an engineering perspective if it does not “break” easily.

Why?

- Reproducibility offers maximum transparency, defends against cherry-picking and builds trust.
- Robustness makes it more likely for a model to survive transition from development to production.

Reproducibility and robustness

Why is it hard?

- Technologically, we need to “code up” every manual step and “codify” every assumption.
- Statistically, we need to control the variance of our answer.
- Ethically, we need to be transparent about every failed experiment and every decision we took.

Is it worth my time learning about this?

- A completely reproducible data science pipeline is a solid foundation for ethical data science work.
- Recent years have seen an explosion in pipeline frameworks, rendering it essential knowledge.

Reproducibility and robustness

Raw data

Exploratory data analysis

Decision on unit of analysis

Creation of master table

Enrich data

Trained model

Reporting

Deployable model

Tracking

Diagnosing

Updating

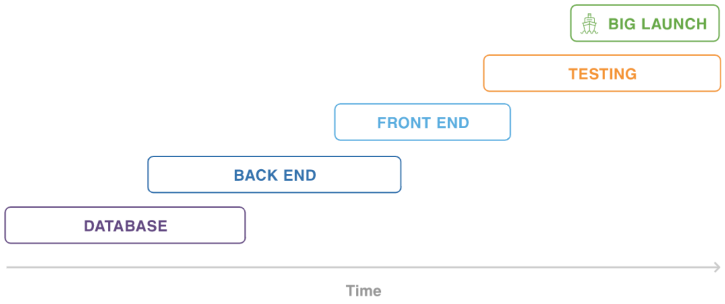
← DataOps →

← MLOps →

← LiveOps →

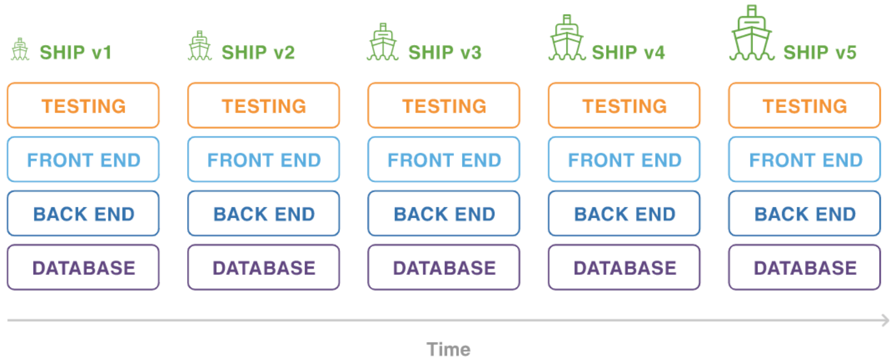
Project philosophies in software development

Waterfall



Clearly defined linear, sequential steps. Revisiting earlier steps hard

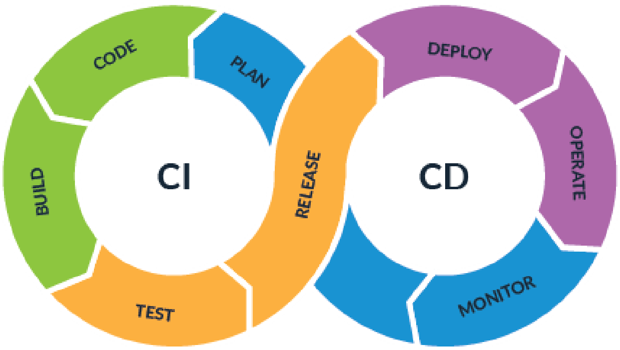
Agile



Iterative process, shipping parts of functionality as soon as they can be ready, collecting user feedback

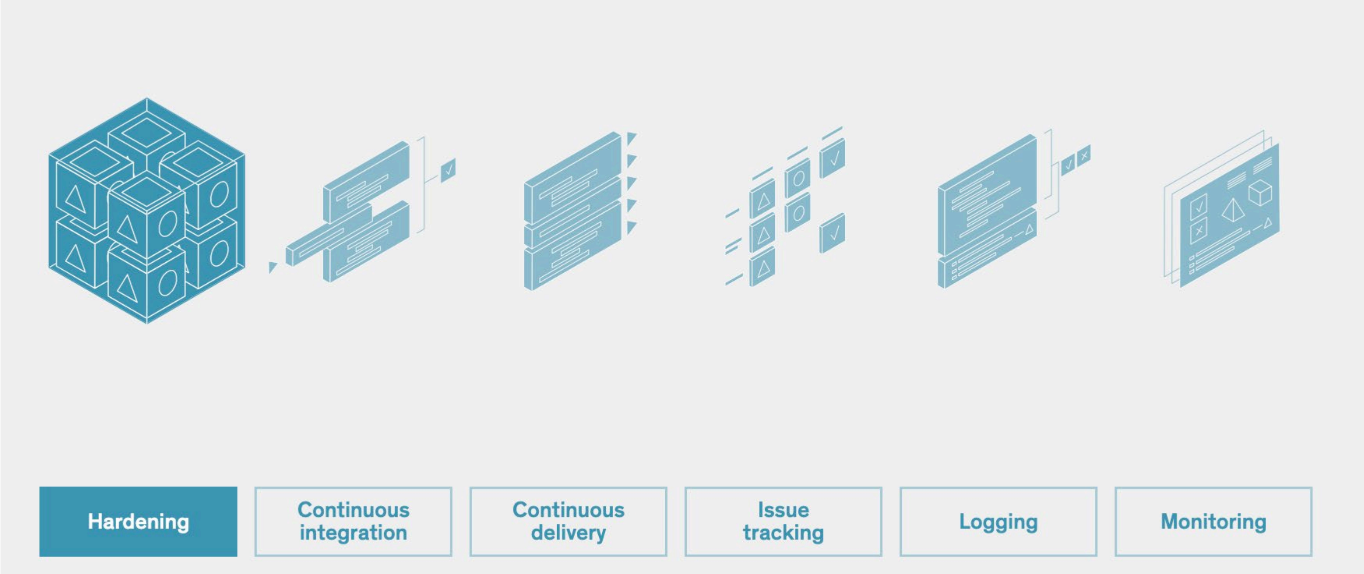
Source of images: https://www.atlassian.com/agile/

CI/CD is core to modern software engineering



- Continuous Integration (CI) allows new code / functionality to be automatically integrated with the main codebase and automatically tested to ensure ”nothing breaks” and standards are met.
- Continuous Delivery (CD) makes it possible to rapidly release and deploy latest version of software in production, relying on cloud tools (e.g., use of containers can help with dependencies)

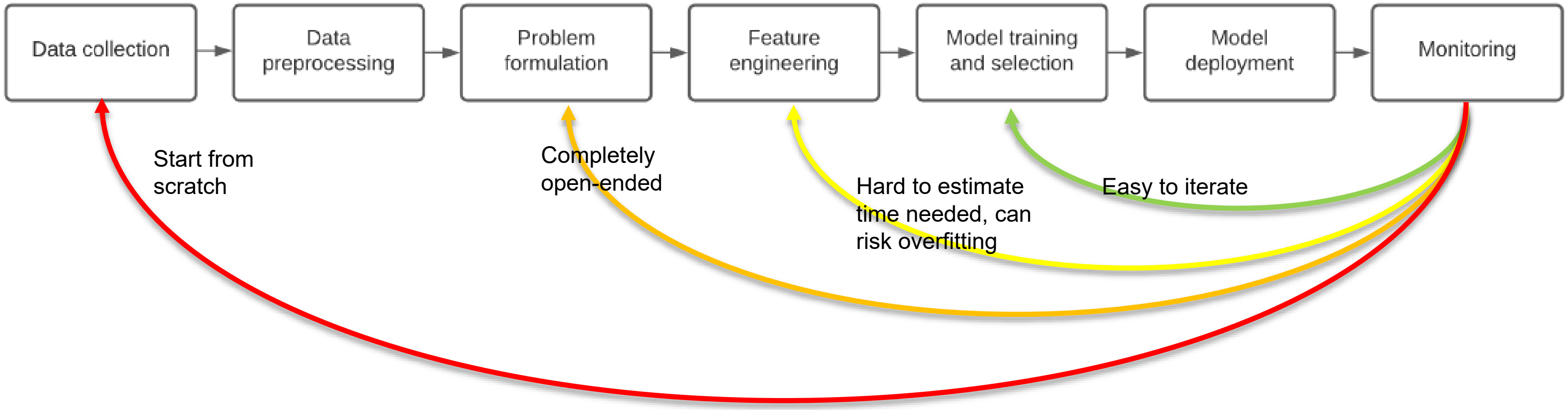
Source: www.analyticsvidhya.com



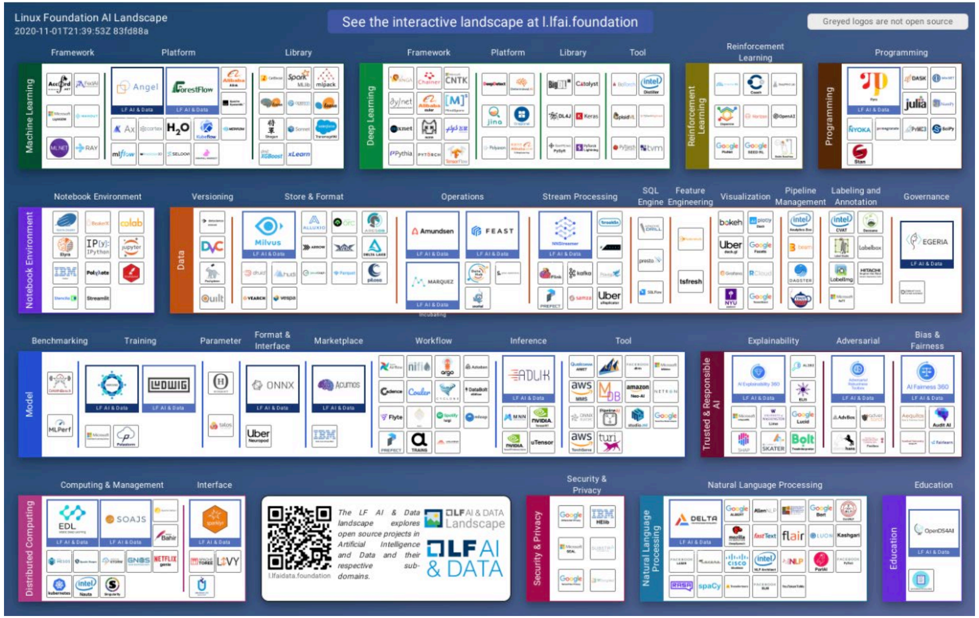
- Automation and the requirement to be able to “roll back” to an earlier version implies reproducibility of results. The frequency at which models are expected to change and still be “shipped to production” implies robustness, or “hardening”, which also involves an InfoSec angle (out of scope).

Source: https://www.mckinsey.com/capabilities/quantumblack/our-insights/executives-guide-to-developing-ai-at-scale#devops/hardening

Reproducibility and robustness



Unlike in most modern software development, it is not always easy to be agile in ML development.

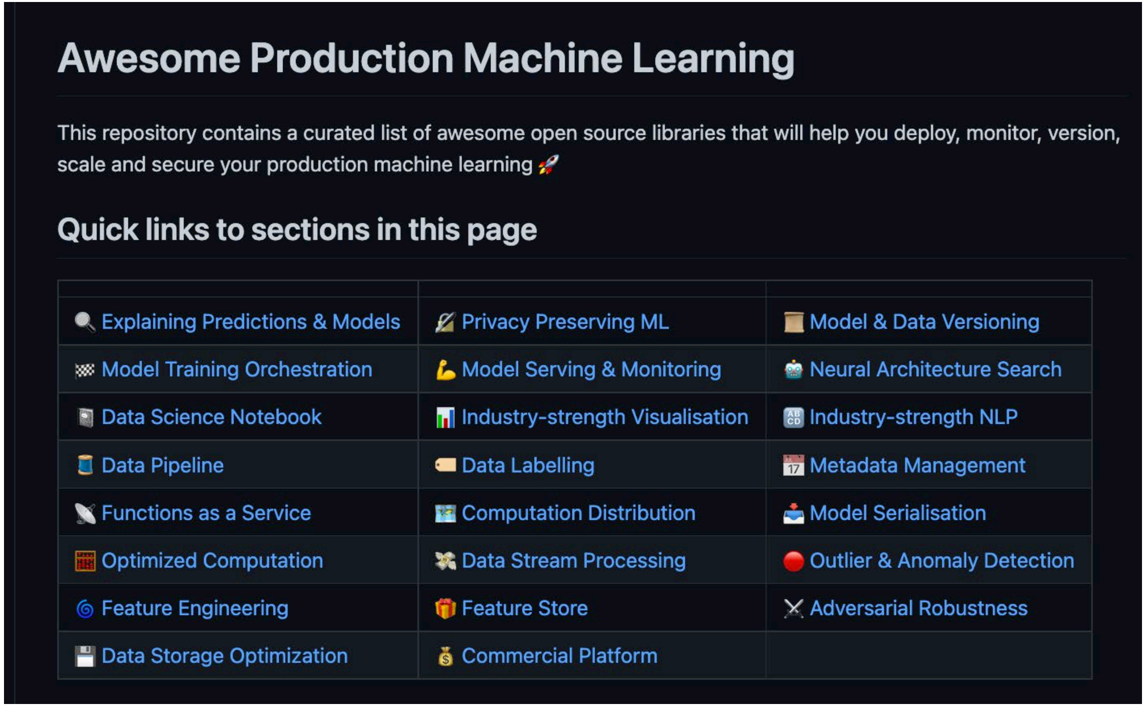


https://landscape.lfai.foundation/

- AutoML
- CI/CD for Machine Learning
- Cron Job Monitoring
- Data Catalog
- Data Enrichment
- Data Exploration
- Data Management
- Data Processing
- Data Validation
- Data Visualization
- Feature Engineering
- Feature Store
- Hyperparameter Tuning
- Knowledge Sharing
- Machine Learning Platform
- Model Fairness and Privacy
- Model Interpretability
- Model Lifecycle
- Model Serving
- Model Testing & Validation
- Optimization Tools
- Simplification Tools
- Visual Analysis and Debugging
- Workflow Tools

https://github.com/kelvins/awesome-mlops

As a result, a huge variety of tools have sprung up to support ML workflows



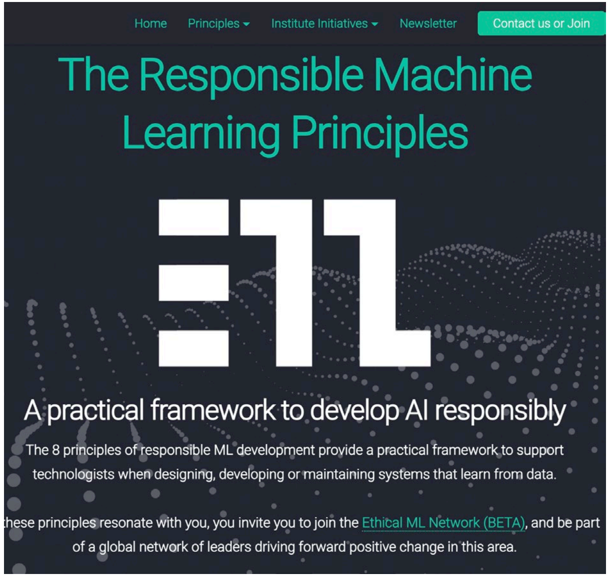
Awesome Production Machine Learning

This repository contains a curated list of awesome open source libraries that will help you deploy, monitor, version, scale and secure your production machine learning 🚀

Quick links to sections in this page

🔍 Explaining Predictions & Models	🔏 Privacy Preserving ML	📜 Model & Data Versioning
🏁 Model Training Orchestration	💪 Model Serving & Monitoring	🤖 Neural Architecture Search
📓 Data Science Notebook	📊 Industry-strength Visualisation	🔠 Industry-strength NLP
🧵 Data Pipeline	🏷️ Data Labelling	📅 Metadata Management
📡 Functions as a Service	🗺️ Computation Distribution	📥 Model Serialisation
🧮 Optimized Computation	💸 Data Stream Processing	🔴 Outlier & Anomaly Detection
🌀 Feature Engineering	🎁 Feature Store	⚔️ Adversarial Robustness
💾 Data Storage Optimization	💰 Commercial Platform	

https://github.com/EthicalML/awesome-production-machine-learning



https://ethical.institute/principles.html

Summary and next steps

- To offer transparency and safety, ML pipelines need to be completely reproducible and robust.
- Modern software development principles such as Agile and CI/CD help us in that direction.
- However, ML development is different to software development and requires bespoke tooling.
- This space has exploded in recent years, and ML in industry is critically reliant on MLOps tooling.
- We start with the basics of creating a reproducible, robust DS pipeline.