Machine Learning and AI

Jour 2 : Développement et Intégration de Projets IA

Jeff Abrahamson

July 2024

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Doing Data Science

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Perspectives

- Data science is iterative
- Start simple, get better

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Perspectives

- Data science is iterative
- Start simple, get better

Sometimes there's no business case to do more.
You want to know where that threshold is so you can stop.

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Risks

- Nothing is guaranteed, but competitors are innovating and experimenting
- Examples from past projects can help, but often leads to “this is different”

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Success

Think early about how to measure success.

Success

Think early about how to measure success.

The definition of success will evolve.

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Andrew Ng Methodology — Lifecycle of an ML Project

Four phases:

1. Project scope definition
2. Data management
3. Modeling
4. Deployment

Each phase has its own particular challenges.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

1. Project Scope Definition

Keypoints:

- Delimitation of the task to be accomplished
- Definition of success indicators
- Budget in terms of time, personnel, etc.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

2. Data Management

Keypoints:

- Unbiased data collection method
- Clear definition of inputs and outputs
- Robust data processing pipeline without training/production disparities
- Reproducibility and experiment management

Hugo Mougard, Machine Learning, CC BY-SA 4.0

3. Modeling

Keypoints:

- Training that takes into account production needs (model size, speed, etc.)
- Error analysis, often by significant data slices
- Reproducibility and experiment management

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

4. Deployment

Keypoints:

- Scaling
- Detection of data and concept drifts
- Monitoring
- Retraining process

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

CRISP-DM

The cross-industry standard process for data mining.

The major phases:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Technical standpoint — MLOps

From a technical standpoint, Data Science projects management is called MLOps:

- Tools bridging the gap from proof of concept to production
- Techniques to comply with regulatory obligations
- Methods for mitigating ethical issues

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

State of MLOps

It's still a young practice, even if it's grown enormously in ten years.

- Rapid evolution, many competing libraries
- Much less stable than DevOps
- Adoption much less homogeneous than DevOps
- Co-evolution with legislation and regulation

Given the instability of the domain, broad principles are more important than specific techniques.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Data

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration *(what's needed, how collected, accuracy)*
- Retrieval, labeling and organisation *(extraction, labeling, organising)*

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Definition:

- Identify Data Requirements
- Data Sources
- Data Standards *(formats, naming conventions, . . .)*

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Calibration

- Data Quality Metrics *(accuracy, completeness, consistency, timeliness, . . .)*
- Validation Rules
- Data Integration
- Tools and Technologies

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Retrieval:

- Data Extraction
- Data Aggregation

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Labeling:

- Metadata assignment
- Data Tagging
- Annotations

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Data Management Phases

Two distinct phases:

- Definition and calibration
- Retrieval, labeling and organisation

Organisation:

- Data Structuring
- Data Storage
- Data Indexing
- Data Management

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Objectives

We want primarily to address two questions:

- What are the relevant inputs and outputs?
- What level of performance can we expect?

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Trash in Trash out Principle

Fundamental Principle of Machine Learning:

Data definition is the *central* point of a machine learning system.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Research vs Industry

$$\text{Research} \quad \text{System} = \text{Data} + \overbrace{\text{Parameters} + \text{Model}}^{\text{Work}}$$

$$\text{Industry} \quad \text{System} = \overbrace{\text{Data} + \text{Parameters}}^{\text{Work}} + \text{Model}$$

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Definition Example – Translation

How to define the output?

1. I was overwhelmed with joy. $\rightarrow$ J'ai été submergé par la joie.
2. I was overwhelmed with joy. $\rightarrow$ Je fus submergé de joie.
3. I was overwhelmed with joy. $\rightarrow$ Je fus terrassé par la joie.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Definition Example – Audio

How to define the input?

1. Um. . . I'll be there in 5 minutes
2. Um, I'll be there in 5 minutes
3. I'll be there in 5 minutes

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Definition Example – Identity Fusion

How to define the output?

- Martin Durant, 44000, . . . , <martin@durant.fr>
- Martin Durant, 44000, . . . , <martin.durant@gmail.com>

Hugo Mougard, Machine Learning, CC BY-SA 4.0

What is at stake?

All these decisions change the function that the model will approximate.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Types of Data

Two primary criteria:

- Size of the dataset
- Structured or unstructured data

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Size of the Dataset

How much data we have influences what is important.

Small Quality of annotations is crucial

Large Quality of data processing processes is crucial

A subset of a large dataset can behave like a small dataset (especially a critical slice).

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Structured / Unstructured Data

Data structure affects what's easy and hard.

Structured Hard to annotate for humans. Hard to augment.

Unstructured Likely easy to annotate for humans. Often augmentable.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Annotation Guide

For annotators:

- Should be as robust as possible
- Ideally written by a mix of domain experts / ML
- Written iteratively:
  - Write a version
  - Annote
  - Detect ambiguous points
  - Write a new version

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Coverage of Input Data

Data coverage principles:

- All cases to be handled must be represented in the data
- All cases to be handled must be represented in sufficient quantities
- It’s particularly important to avoid discrimination on protected attributes

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Calibration

Estimating expected performance:

- Bibliographic research on existing approaches
- Estimation of human performance

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Human Level Performance (HLP)

Human Level Performance (HLP) refers to the capability of AI systems or models to perform tasks at a level comparable to that of a human.

Achieving HLP means that the AI can handle specific tasks with a similar degree of accuracy, efficiency, and reliability as a human expert in that domain.

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Human Level Performance

- **Benchmark for Performance:** HLP serves as a benchmark to estimate the potential maximum performance of a system.
- **Bayes' Error Estimation:** It helps in estimating the Bayes' error, which is the minimum possible error due to the inherent randomness in the data.
- **Annotation by External Processes:** HLP is crucial when annotations are generated by external processes rather than human annotators.
- **Unstructured Data:** Particularly relevant for tasks involving unstructured data (e.g., images, audio, text).
- **Achievable Performance:** Provides an idea of the best possible performance that can be achieved by a system.

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Improving Human Level Performance

- **Underestimating HLP:** Sometimes, HLP is underestimated to make it easier for AI models to surpass human performance.
- **Impact on Orientation:** Poorly defined HLP can mislead the direction of development efforts, resulting in suboptimal performance improvements.

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Improving Human Level Performance — Example

Example of HLP in Action:

1. "Um. . . I'll be there in 5 minutes."
2. "I'll be there in 5 minutes."

- If 80% of annotators prefer the first transcription and 20% prefer the second, the agreement between two random annotators is calculated as: $0.8^2 + 0.2^2 = 68\%$ agreement.
- An algorithm that always chooses the most common transcription can achieve 80% agreement.
- Significant errors may be obscured by these seemingly high agreement rates, highlighting the need for deeper analysis beyond superficial gains.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Documentation

Things to track during the definition and calibration process:

- Potential biases in the data that you suspect
- Real data coverage issues
- Regulatory issues related to the data

This information is crucial for properly documenting the model in the long run.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Data Processing

Data Organization

Many options:

- Structuring (schema, description, ...)
- Scaling (SQL, NoSQL, distributed file system, ...)

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Feature Store

In an MLOps context, feature stores are often an intermediate step between the original source and processing.

- Allows storage optimized for ML
- Avoids recomputing the same features
- Enables discoverability

See *Feast* for example.
`https://docs.feast.dev/`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Data Acquisition

- Aim for a short first iteration to get feedback for subsequent phases
- List potential sources and their time/money budget
- Potentially have the ML team do initial annotation
- Define competent profiles

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Iterations on Data

- Order of magnitude: not more than x10 at once
- Work jointly on source quality, processes, volume
- Very different in prototyping and production phases:

Prototyping	Gathering enough data to decide go/no-go
Production	Deep work as the main vector to reach performance ceiling

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Metadata

Because data is the heart of an ML system, follow good practices.

- Track provenance
- Track transformations
- Maintain reproducibility of data acquisition and transformations:
  - Increasingly important for regulation
  - Essential for debugging

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Metadata — Consequences

Think about preserving metadata.

- Define data acquisition processes
- Define data transformations
- Implement dedicated systems

This leads to many architecture decisions.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Processing Pipelines

- Provide reproducibility and process automation
- Often are directed acyclic graphs of operations
- Many solutions exist

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Desirable Characteristics

Strengths of a processing pipeline

- Adaptation to batch (development) as well as real-time (production)
- Faithfulness of development/production processing
- Scalability
- Deployment on various targets
- Ease of development
- Integration with the ecosystem

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Main Processing Pipeline Solutions

Strong points of each solution:

Deployment options, dev/prod parity, performance
`https://www.tensorflow.org/tfx`

Kubernetes integration `https://www.kubeflow.org/`

Reproducibility and “low tech” solution `https://dvc.org/`

Flexible, easy to adopt, supported by clouds
`https://mlflow.org/`

Kubernetes integration `https://www.pachyderm.com/`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Modeling

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Model Engineering

Several key aspects to consider:

- Performance in production
- Various deployments with very distinct characteristics
- Interpretability
- Maintainability
- Compliance with regulations (non-discrimination, . . .)

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Best Practices

In an industrial context, the focus is on the data, not the model :

- Start simple (heuristic, simple model)
- Use the industry standard for the task given your performance and deployment requirements
- Improve data first, model second

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Model Card

Model cards were introduced in *Model Cards for Model Reporting*
`https://arxiv.org/abs/1810.03993`

- Specifies ethical and regulatory decisions related to the model
- Provides transparency
- Goal is informing both the public and internal teams

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Model Registry

Centralized place to store and retrieve models and associated meta-data :

- Eases deployment
- Solidifies reproducibility

`https://mlflow.org/docs/latest/model-registry.html`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Models and Error Analysis

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Introduction

Error analysis is crucial during development:

- Guides future work
- Determines potential progress

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Link between error analysis and interpretability

- Interpretability enables error analysis
- Transparency (sometimes a regulatory or functional requirement in production)
- Interpretability-performance continuum:
  - Interpretable models often insufficient to approximate desired functions
  - Modern performant models are black boxes

Often a dilemma in choosing a model.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Link between error analysis and interpretability

- Interpretability enables error analysis
- Transparency (sometimes a regulatory or functional requirement in production)
- Interpretability-performance continuum:
  - Interpretable models often insufficient to approximate desired functions
  - Modern performant models are black boxes

Often a dilemma in choosing a model.

Useless tip: humans are often not explainable.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Recommended Approach

Some heuristics for error analysis:

- Determine relevant data slices
- Estimate model performance and achievable performance on each slice
- Prioritize work on slices offering the most impact

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Data Slices

Idea popularized by Apple in their paper *Overton: A Data System for Monitoring and Improving Machine-Learned Products* and *Snorkel* `https://www.snorkel.org/`.

Excellent Snorkel blog on the subject. `https://www.snorkel.org/blog/slicing`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Recommended Approach — Example

Let's consider working on an image classification system.

Suppose we distinguish the following slices in our image data:

- Presence of mountains
- Presence of humans
- Presence of cars

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Recommended Approach — Example, Continued

We estimate the following performances:

Slice	HLP	Model
Mountains	65%	60%
Humans	96%	90%
Cars	80%	40%

Which slice is most important for improvements in the next iteration?

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Recommended Approach — Example, Conclusion

Using proportions in the dataset to quantify impact:

Slice	HLP	Model	Proportion	Potential Impact
Mountains	65%	60%	10%	0.5%
Humans	96%	90%	85%	5.1%
Cars	80%	40%	1%	0.4%

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Error Analysis

Once relevant slices are identified, some options:

- Find explanatory examples of the model
- Use a simpler model that globally or locally explains the complex model

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Explanatory Examples

Things to look for:

- **Prototypes** Representative examples of model behavior
- **Counterfactual examplqes** Modification of existing instances to see how prediction evolves
- **Adversarial examples** Counterfactual examples with a significant impact on prediction
- **Influential examples** Examples that have had the most impact on the model

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Explanatory Models

Training models:

- Simple
- Interpretable (linear regression, simple trees, etc.)
- Approximating the complex model
- Helping to understand important features
- Globally (multiple examples, broad coverage)
- Or locally (one example)

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Measures

Once the analysis is done:

- Data augmentation
- Feature engineering
- Exploration of new parameters

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Further Reading

Interpretable Machine Learning book
`https://christophm.github.io/interpretable-ml-book/`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Intro to Production Deployment

Deployment Challenges

- Providing a trained model to a diverse and large crowd
- With ML performance observed in quality tests
- With good classical performance (latency, throughput, . . .)

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Common Pitfalls

What can go wrong? *(Or: what often goes wrong?)*

- Different code in development vs production
- Poor dependency management
- Inadequate or inappropriate architecture

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Deployment Best Practices

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Deployment Strategies

Several criteria allow for choosing a deployment strategy:

- Service traffic
- Audience (public, internal, other services, . . .)
- Deployment frequency
- . . .

Key concepts:

- Progressive deployment
- Rollback (ability to return to the previous state of the system)

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Shadow Deployment

- Deployment of the model alongside the existing system
- Model outputs are not used by the application
- Analysis of model outputs and decision to continue deployment or not

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Blue/Green Deployment

- Deployment of the new model (green) alongside the existing system (blue)
- When tests are successful on the green system, traffic is redirected there
- Allows for rollback if issues arise

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Canary Deployment

- Similar to blue/green, two parallel systems
- Gradual ramp-up of the green system

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Implementation

The two most popular options:

- Kubernetes + istio
- Internal tooling

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

CI/CD

The launch of these deployments can be done via git tags and operations in CI/CD.

This is the gitops approach with gto project from DVC, for example

`https://www.atlassian.com/git/tutorials/gitops`
`https://dvc.org/doc/gto`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Monitoring

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Challenges

- Model drift
- Ensuring proper behavior of a model
- Knowing when to retrain a model
- Understanding the data of a model

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Common Issues

Data Drift: the evolution of the input data distribution.

Gradually makes models obsolete, so need to retrain.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Common Issues

Concept Drift: the evolution of the correspondence between outputs and inputs.

Gradually makes models obsolete, so need to retrain.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Speed of Data Evolution

Different data sets evolve at different rates.

- **User Data** often changes slowly but fundamentally
- **Enterprise Data** often changes more markedly and abruptly

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Common Issues

Data Quality:

- Missing values
- Outliers
- Schema changes
- ...

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Tools — Test Suites

Automated execution for detection of all these challenges:

- Data drift
- Concept drift
- Poor data quality
- Drop in prediction performance

The most widely used library for this is `evidently`.

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Tools — Dashboard

The human-facing counterpart of test suites:

- To fuel discussion on data within the Machine Learning team
- To complement more general dashboards

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Classic Monitoring

In addition to these Machine Learning-specific issues, a production system needs usual monitoring.

E.g., the four "golden signals" in Site Reliability Engineering:

- Latency
- Traffic
- Errors
- Saturation

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Copyright 2024 Jeff Abrahamson, CC BY-SA 4.0

Implementation

Several solutions are common:

- Evidently `https://www.evidentlyai.com/`
- Seldon Core `https://www.seldon.io/solutions/core-plus`
- TFX ExampleValidator `https://www.tensorflow.org/tfx/guide/exampleval`
- Great Expectations `https://docs.greatexpectations.io/docs/home/`

Hugo Mougard, Machine Learning, CC BY-SA 4.0

Case Study

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0

Let’s talk about you

Copyright 2024 Jeff Abrahamson, 🅭🅯🄎 CC BY-SA 4.0