Machine Learning Models to Production

Filipa Peleja
Lead Data Scientist at Levi Strauss & Co Europe
16 November 2021

Machine Learning is not just code, it's code plus data

ML Ops is a set of practices that combines Machine Learning, DevOps and Data Engineering, which aims to deploy and maintain ML systems in production reliably and efficiently

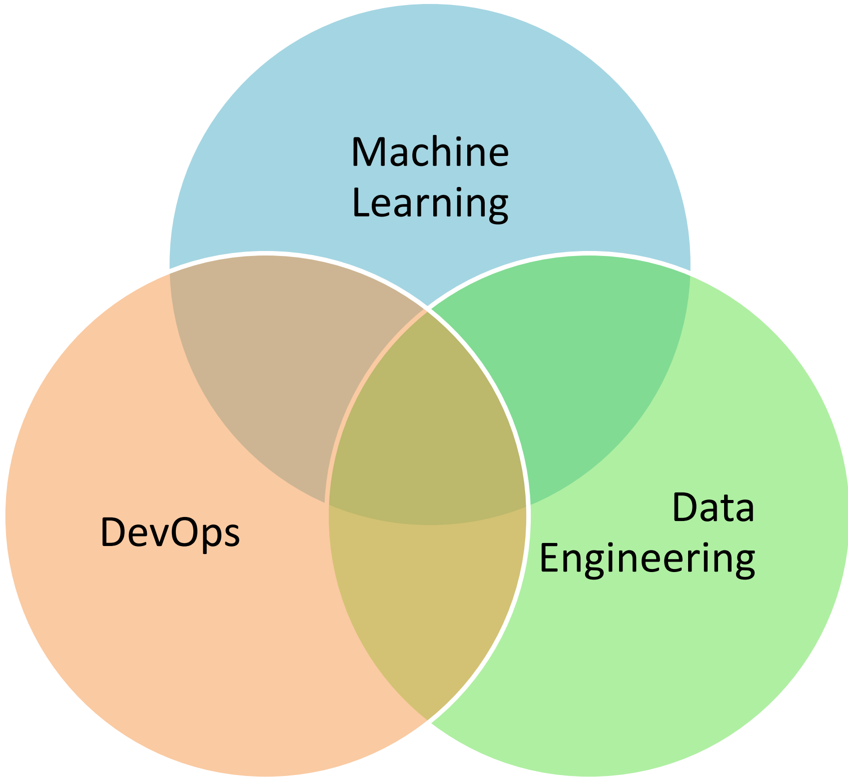
Cristiano Breuel, Machine Learning Engineer

Machine Learning Models in Production

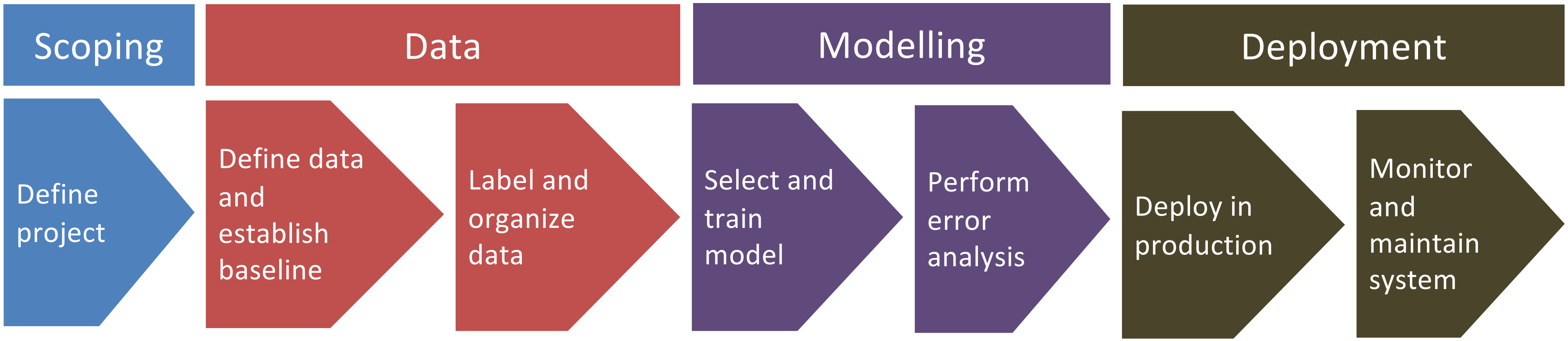
Risks

- Slow, brittle and inconsistent deployment
- Lack of reproducibility
- Performance reduction (training-serving skew)

To avoid such risks we need to combine practices from DevOps, Data Engineering and Machine learning know-how

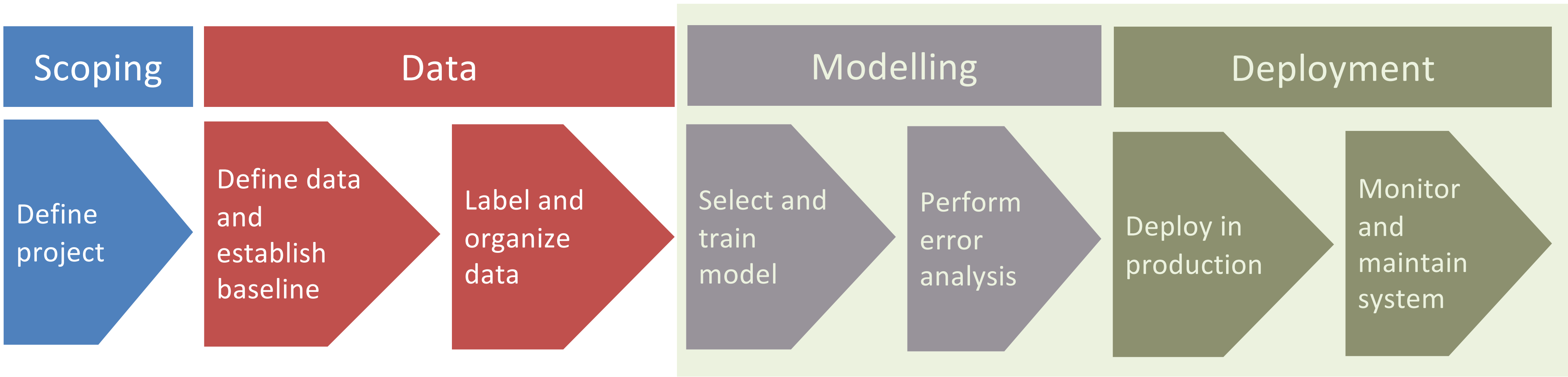


Machine Learning (ML) Lifecycle



Filipa Peleja – filipapeleja@gmail.com

Machine Learning (ML) Lifecycle



Filipa Peleja – filipapeleja@gmail.com

Modelling
Data Centric <> Model Centric

MODEL CENTRIC

improving the model & **data fixed**

vs

DATA CENTRIC

feeding the model high-quality data & **model fixed**

a simpler algorithm with reasonable-quality data will perform *fine* and

will probably outperform
a "better" algorithm that had not-so-good-quality data

Work towards a practical system that works instead of going after latest state-of-the-art algorithm

Filipa Peleja – filipapeleja@gmail.com

Modelling
Model Performance

Deploy a model that performs good on average

but presents <u>poor results on relevant tests</u> is not acceptable

Evaluate the model separately to guarantee

- Overall quality
- Is fully aligned with key tests aligned with stakeholders
- Avoid discriminatory behavior

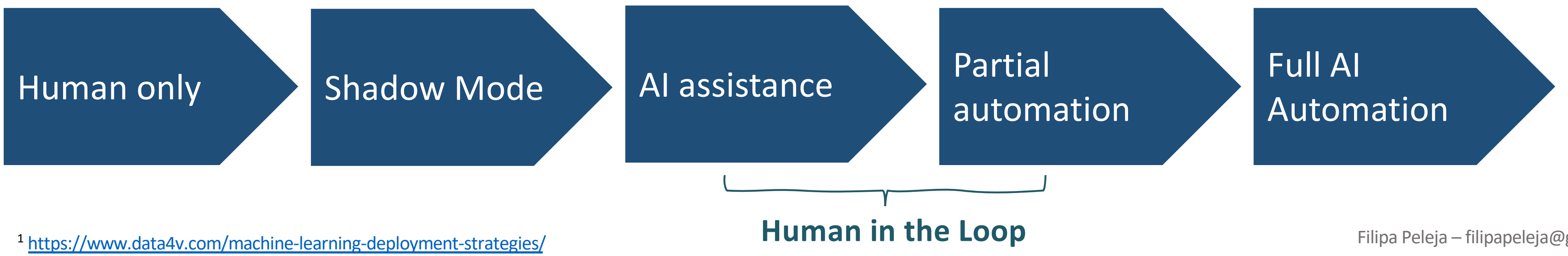
Modelling Deployment

Deployment should not be framed as binary outcome but instead as a **spectrum of varying degrees of automation**

Degrees of automation varies on use case and business alignment

Not always there is a full automation

e.g., partial-automation with a human in the loop might be an AI-based solution for medical diagnostics



[1] https://www.data4v.com/machine-learning-deployment-strategies/

Modelling
Deployment Scenarios

Shadow

How: ML model runs in parallel together with human workflow
Rational: Validate model performance together with humans to ensure the outcome is aligned

Canary

How: Deploy model on a smaller fraction of the target output e.g., 5% email targeting or traffic
Rational: Evaluate the model without exposing it to all targets

Blue-Green

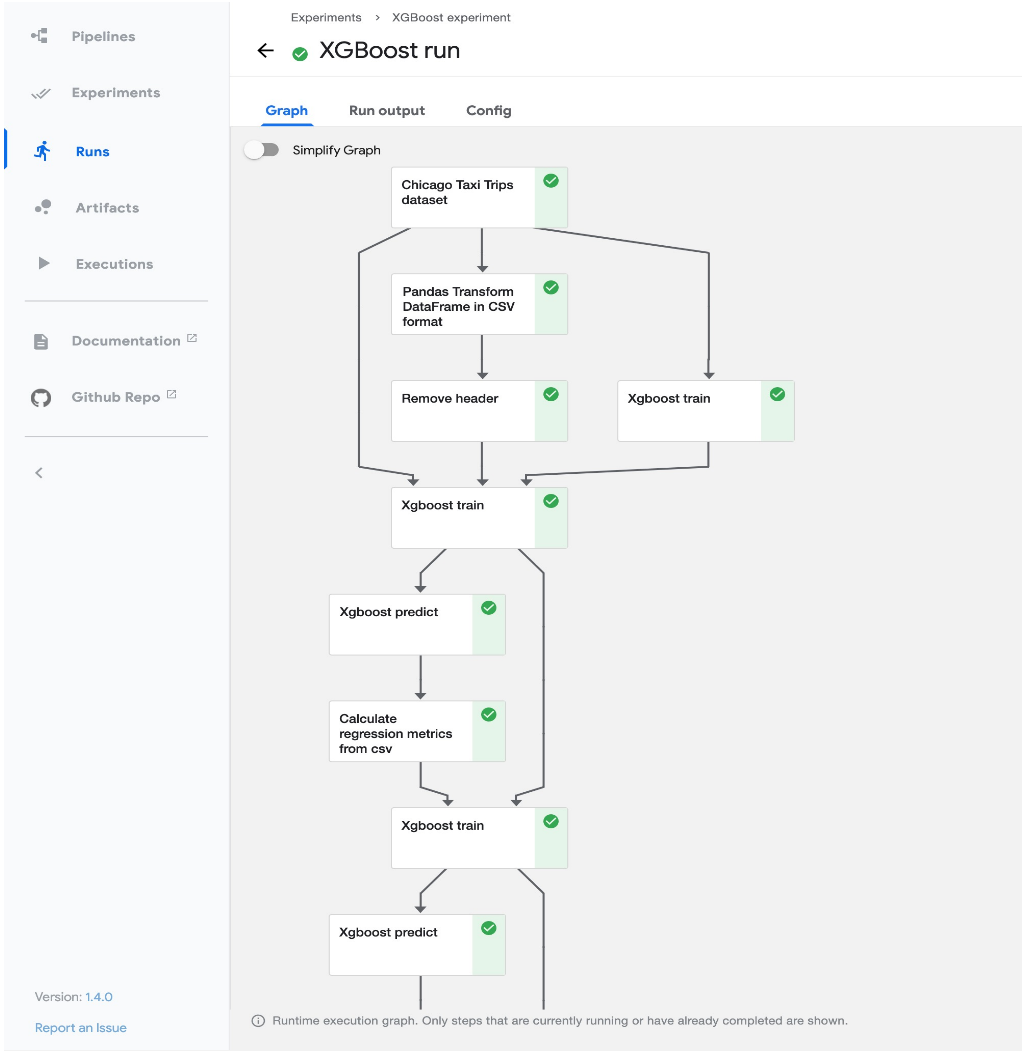
How: replace partially or completely an existing model (blue) with a new model version (green) whereas Blue- and Green- model have nearly identical production environments
Rational: Ensure the downtime to users is minimum and, in case of a problem, roll-back with this deployment strategy

ML Deployment

Visual representation of ML pipeline

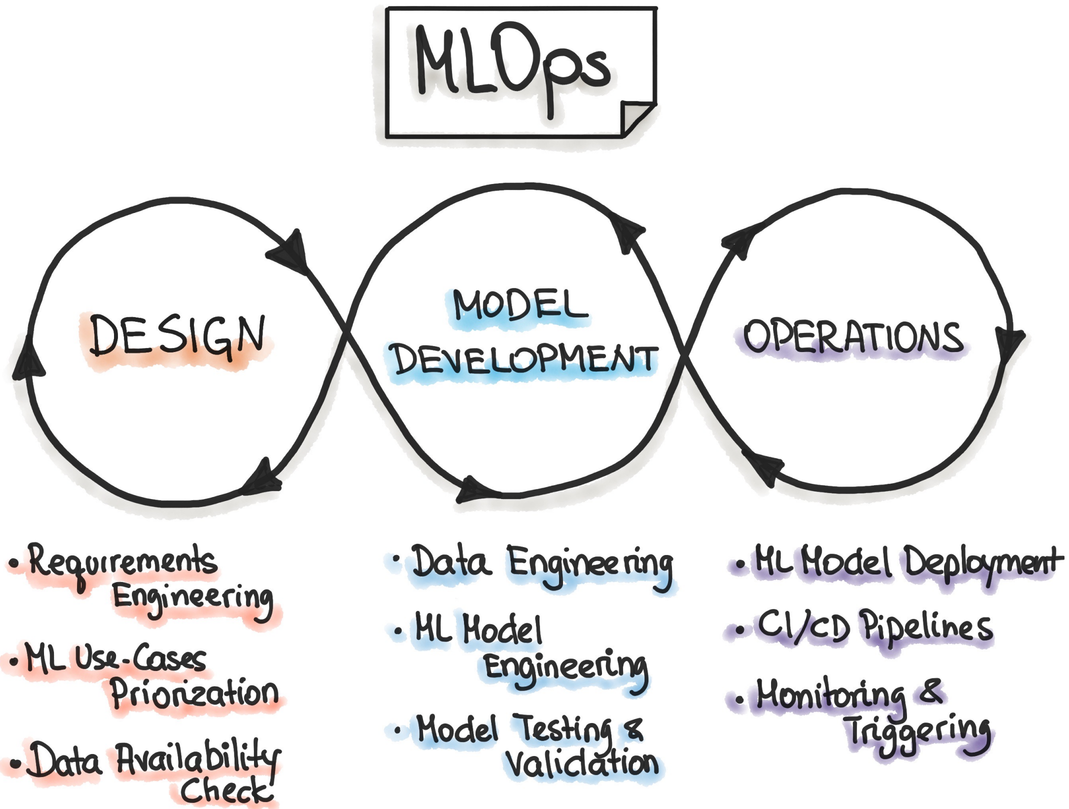
Kubeflow Pipelines

- ML training can have many components
- Data pipeline, training, prediction, deploy..
- In general, needs 2 versions of the pipeline
  - 1 for training and 1 for serving



https://www.kubeflow.org/docs/images/pipelines-xgboost-graph.png

ML Pipeline is pure code artifact which means that is possible to track its versions in source control and automate its deployment using CI/CD[1] pipelines



https://ml-ops.org/content/mlops-principles

[1] Continuous Integration & Continuous Development

Filipa Peleja – filipapeleja@gmail.com

Modelling
System Monitoring

Define metrics to monitor

- Input metrics e.g., data drift, missing values
- Output metrics e.g., model drift, volume target predictions
- Software metrics e.g., server load, latency

Build a dashboard and/or produce automatic alarmistic systems

Modelling
System Monitoring

Define metrics to monitor

- Input metrics e.g., data drift, missing values
- Output metrics e.g., model drift, volume target predictions
- Software metrics e.g., server load, latency

But.. which metrics to track?

Build a dashboard and/or produce automatic alarmistic systems

Model and Data Monitoring

Models and metadata can be tracked using versioning tools (e.g., Git) **but in general data is too large and dynamic/mutable for that to be a realistic option**

ML should be continuous

- Decompose each part of ML pipeline into small, manageable components to be tested and developed separately e.g., processing data and training the model

If the metrics are chosen based on the component purpose the monitoring implementation will become a lot easier

Back to Model Monitoring

In general, ML models do not provide 100% correct results

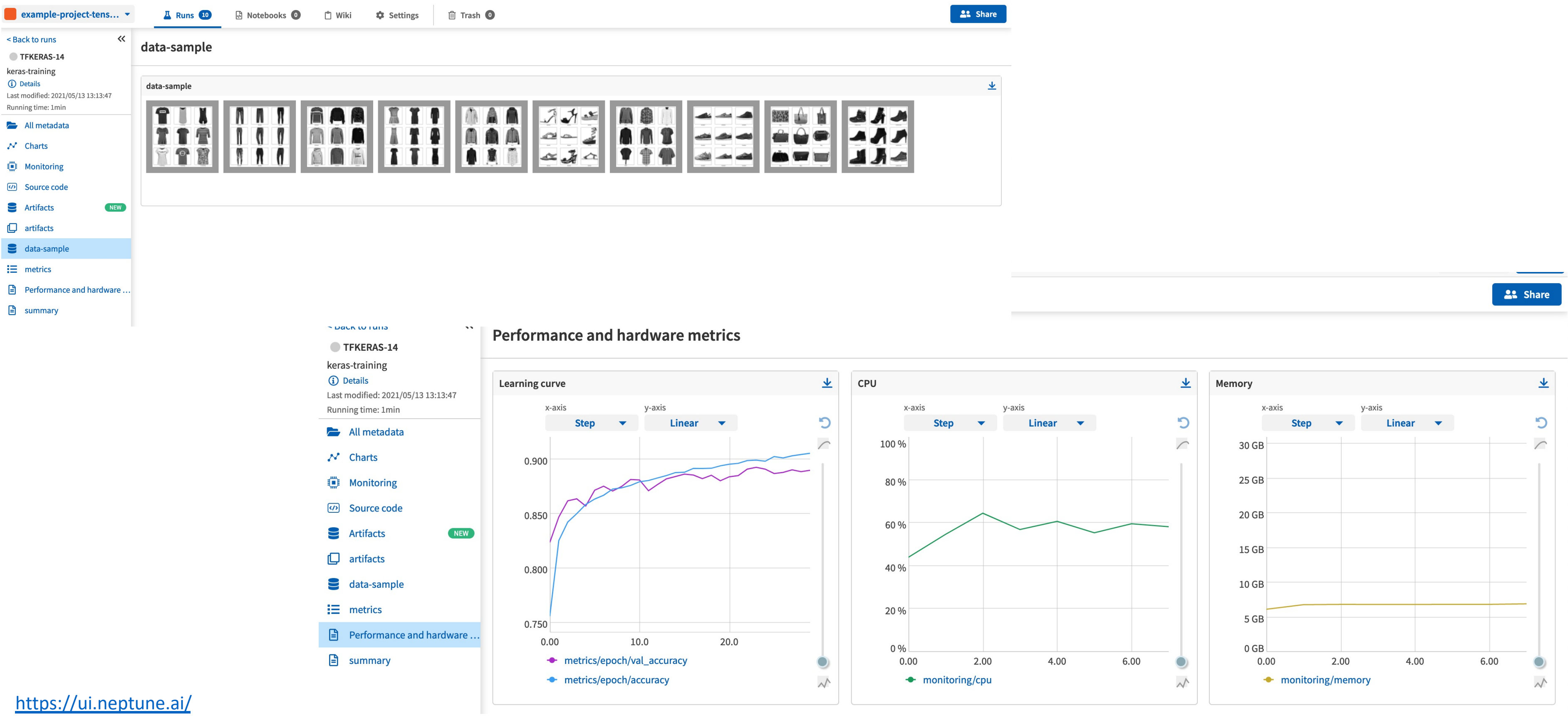
- Model validation should be of statistical nature instead of traditional software development binary pass/fail test

How we decide if the model is good enough for deployment?

- based on the model nature e.g., ensemble models track adjustments to parameters and/or hyperparameters
- decide on thresholds for acceptable values e.g., compare with performance from previous model runs
- sometimes performance cannot be seen as a ”whole” therefore slices of test data are taken to validate model performance for specific relevant features e.g., gender and/or country

Let's have a look at some examples

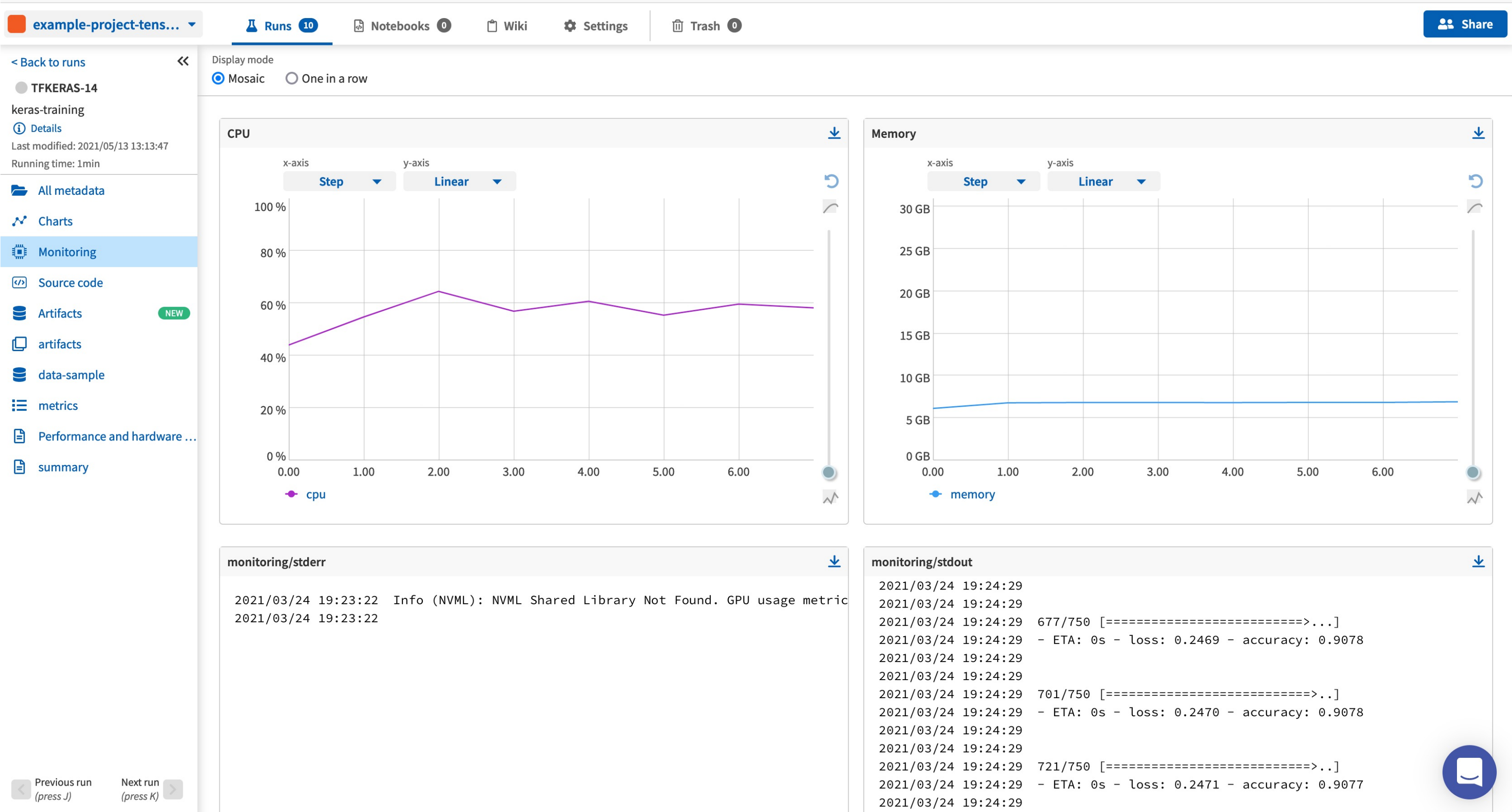
neptune.ai



https://ui.neptune.ai/
Example from app.neptune.ai

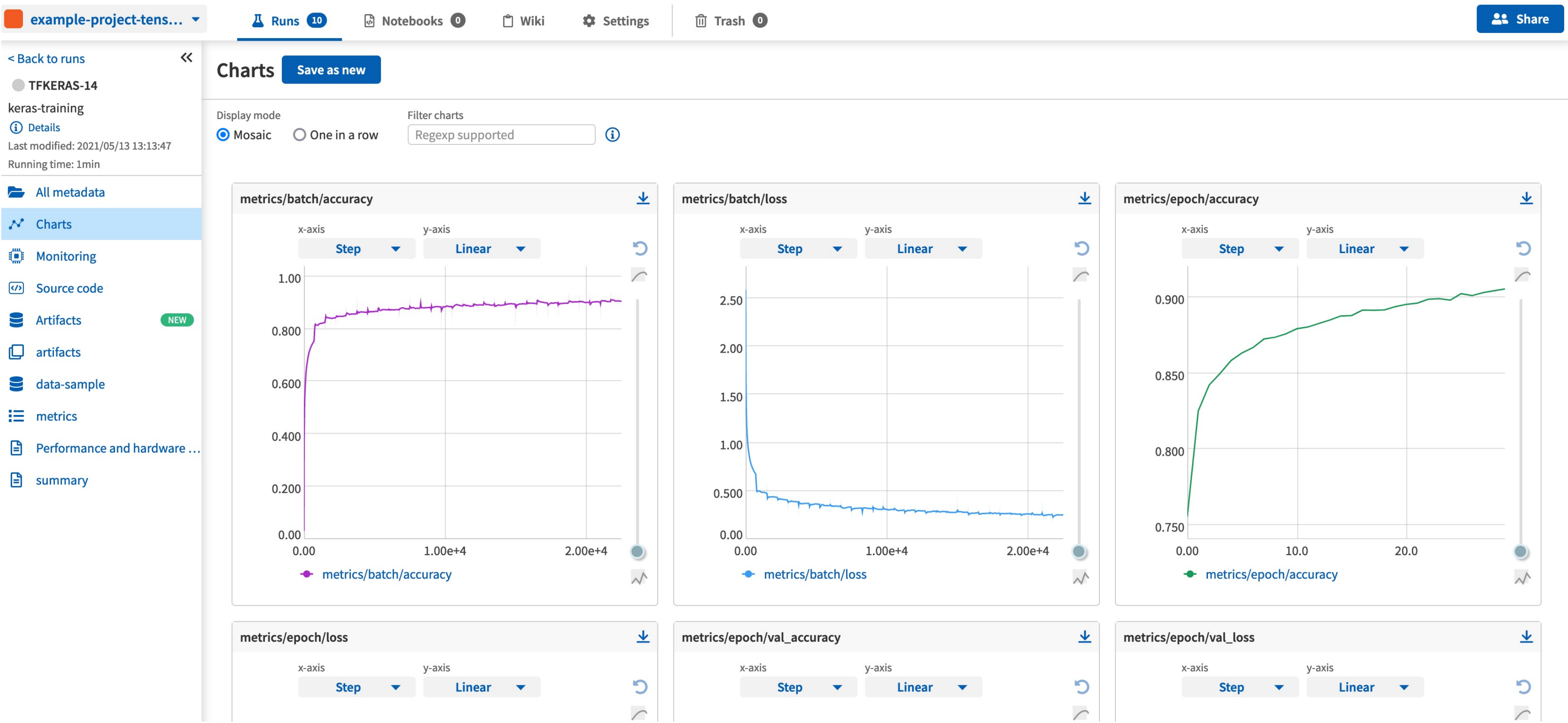
Filipa Peleja – filipapeleja@gmail.com

neptune.ai



https://ui.neptune.ai/
Example from app.neptune.ai

neptune.ai

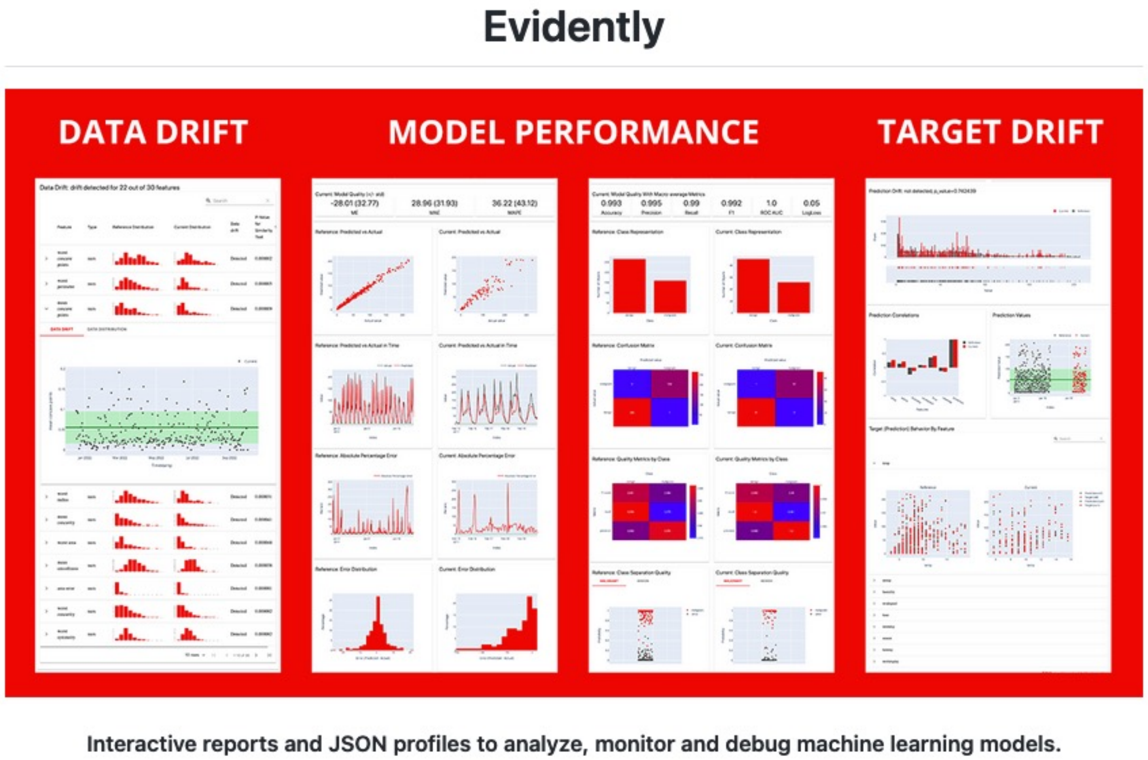


https://ui.neptune.ai/
Example from app.neptune.ai

Filipa Peleja – filipapeleja@gmail.com

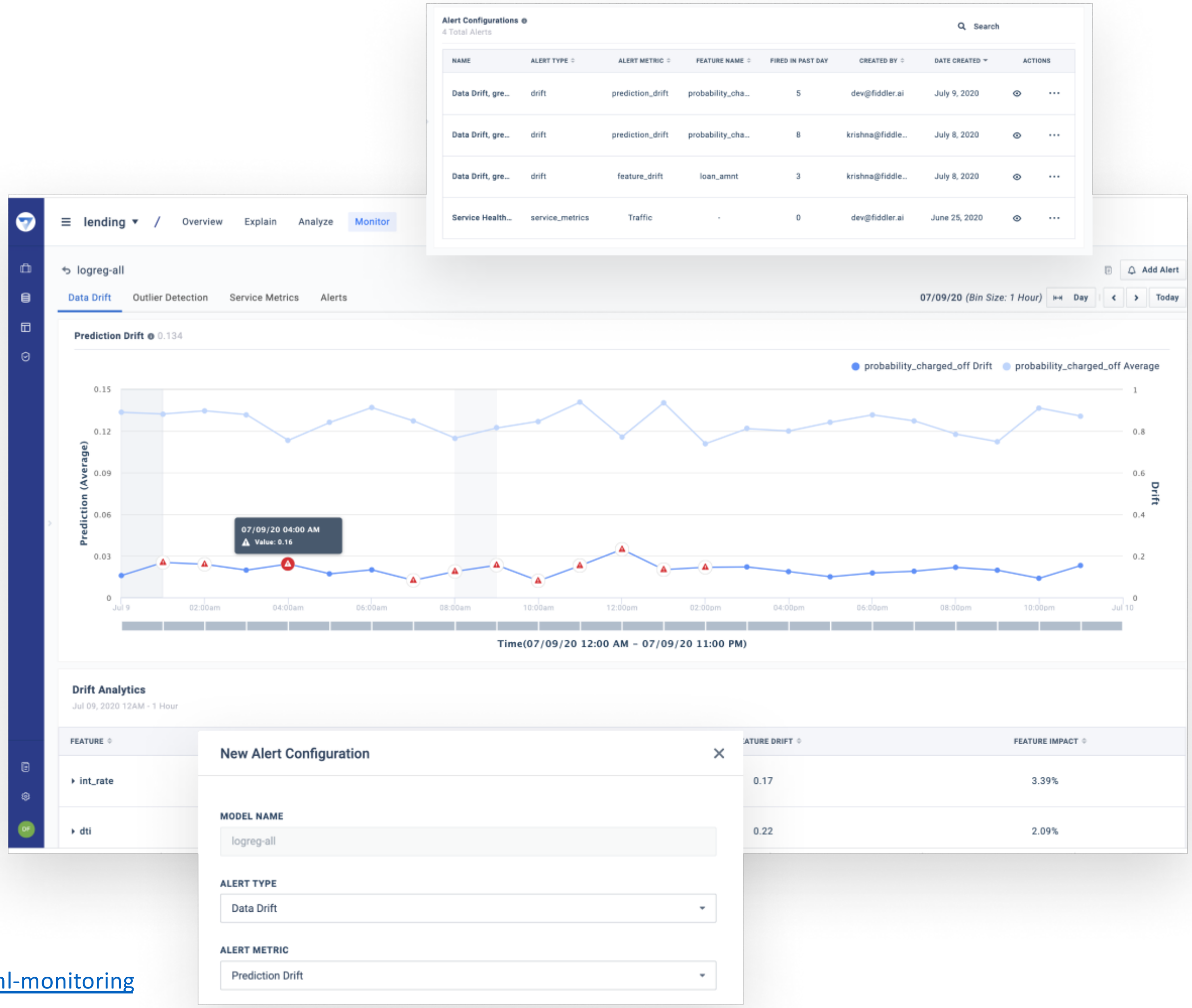
Evidently

Evidently[1] helps evaluate machine learning models during validation and monitor them in production. The tool generates interactive visual reports and JSON profiles from pandas DataFrame or csv files.



[1] https://github.com/evidentlyai/evidently

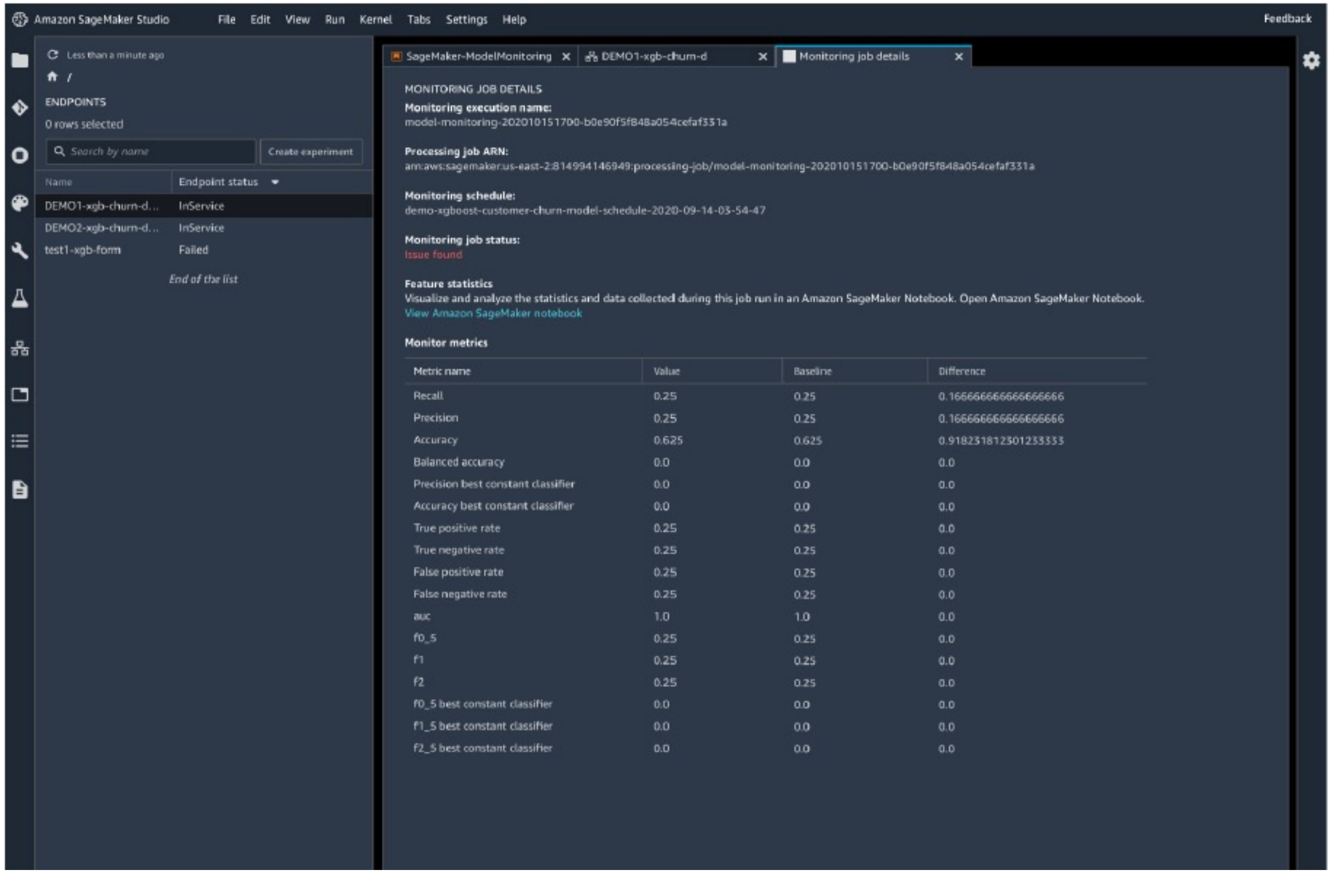
Fiddler



[1] https://www.fiddler.ai/ml-monitoring

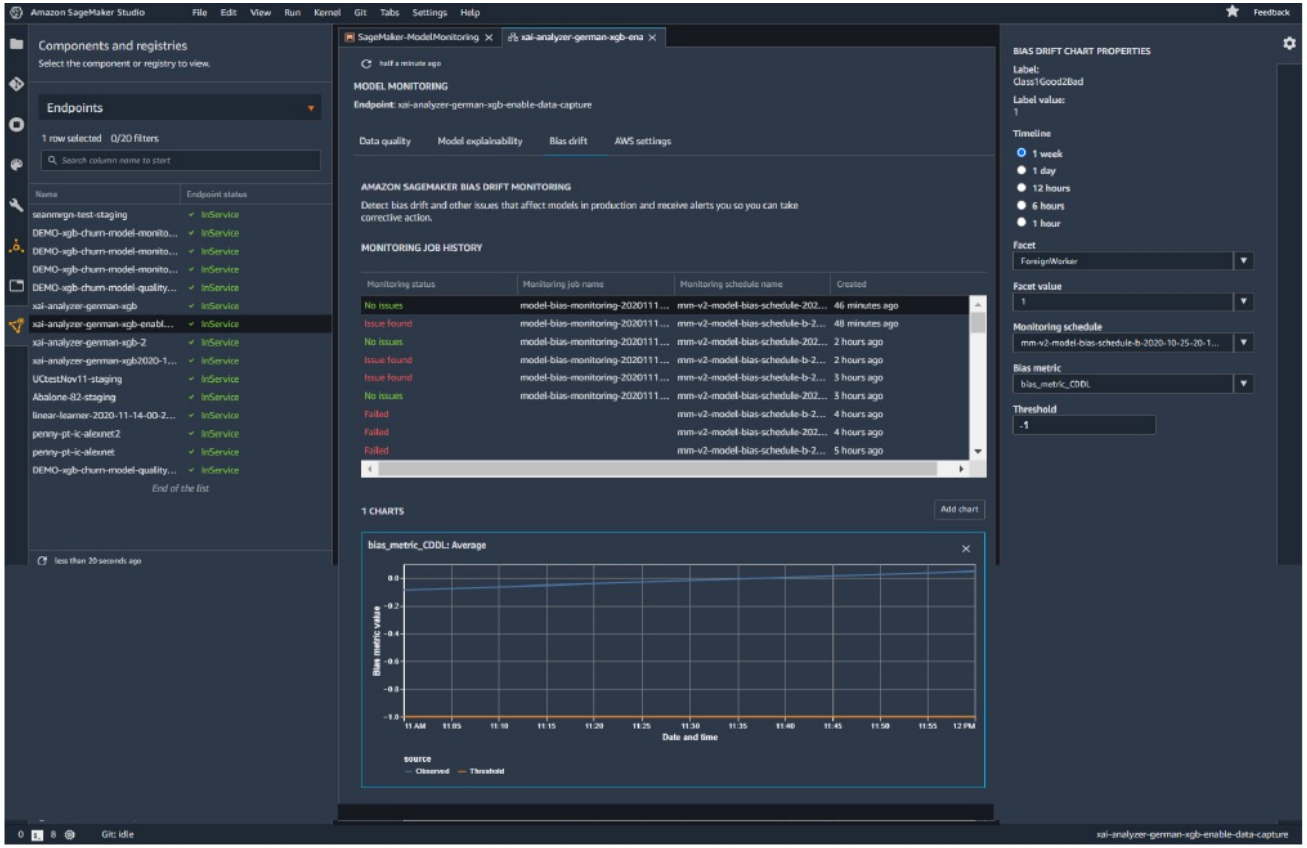
Filipa Peleja – filipapeleja@gmail.com

Amazon SageMaker



[1] https://aws.amazon.com/sagemaker/model-monitor/

Amazon SageMaker



[1] https://aws.amazon.com/sagemaker/model-monitor/

Filipa Peleja – filipapeleja@gmail.com

Recap

Practice	DevOps	Data Engineering	ML Ops
Version control	Code version control	Code version control Data lineage	**Code version control + Data versioning + Model versioning (linked for reproducibility)**
Pipeline	n/a	Data pipeline/ETL	**Training ML Pipeline, Serving ML Pipeline**
Behavior validation	Unit tests	Unit tests	**Model validation**
CI/CD	Deploys code to production	Deploys code to data pipeline	**Deploys code to production + training ML pipeline**
Data validation	n/a	Format and business validation	**Statistical validation**
Monitoring	SLO-based	SLO-based	**SLO + differential monitoring, statistical sliced monitoring**

https://geniusee.com/single-blog/mlops-practices-and-its-benefits

Image from "but Asking Can Hurt," She Writes - Ask Questions

Thank you!

Filipa Peleja

https://www.linkedin.com/in/peleja/