Continuous Intelligence

Moving Machine Learning Application into Production Reliably

Christoph Windheuser
Danilo Sato
Emily Gorcenski
Arif Wider
ThoughtWorks Inc.

WORKSHOP ON WHY AND HOW TO APPLY **CONTINUOUS DELIVERY TO MACHINE LEARNING (CD4ML)**

Strata Data Conference London, April 30, 2019

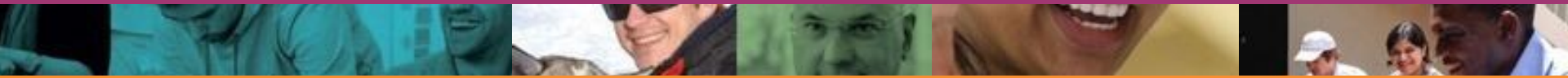
Structure of Today's Workshop

- INTRODUCTION TO THE TOPIC
- EXERCISE 1: SETUP
- EXERCISE 2: DEPLOYMENT PIPELINE
- BREAK
- EXERCISE 3: ML PIPELINE
- EXERCISE 4: TRACKING EXPERIMENTS
- EXERCISE 5: MODEL MONITORING

ThoughtWorks®

5000+ technologists with 40 offices in 14 countries

Partner for technology driven business transformation



join.thoughtworks.com

100+

books written

#1

in Agile and Continuous Delivery

Strata Data Conference London, April 30, 2019

Strata Data Conference London, April 30, 2019

TECHNIQUES

Continuous delivery for machine learning (CD4ML) models #8

ASSESS

CONTINUOUS INTELLIGENCE CYCLE

5. Execute
Changing the real world!

4. Productionize
Planning and prioritizing actions. Hypothesis testing.

1. Acquire
Values attributed to parameters.

2. Store, clean, curate, featurize
Data that has meaning and is fit for consumption and analysis.

3. Model
Understanding, predicting, forecasting and pattern discovery.

PRODUCTIONIZING ML IS HARD

HOW DO WE APPLY DECADES OF SOFTWARE DELIVERY EXPERIENCE TO INTELLIGENT SYSTEMS?

Production systems should be:

- Reproducible
- Testable
- Auditable
- Continuously Improving

CD4ML isn't a technology or a tool; it is a practice and a set of principles. Quality is built into software and improvement is always possible.

But machine learning systems have unique challenges; unlike deterministic software, it is difficult—or impossible—to understand the behavior of data-driven intelligent systems. This poses a huge challenge when it comes to deploying machine learning systems in accordance with CD principles.

PRODUCTIONIZING ML IS HARD

HOW DO WE APPLY DECADES OF SOFTWARE DELIVERY EXPERIENCE TO INTELLIGENT SYSTEMS?

Production systems should be:

- Reproducible
- Testable
- Auditable
- Continuously Improving

Machine Learning is:

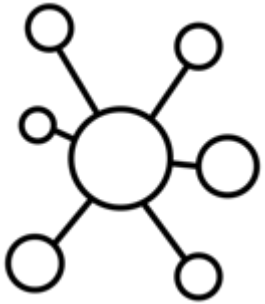
- Non-deterministic
- Hard to test
- Hard to explain
- Hard to improve

CD4ML isn't a technology or a tool; it is a practice and a set of principles. Quality is built into software and improvement is always possible.

But machine learning systems have unique challenges; unlike deterministic software, it is difficult—or impossible—to understand the behavior of data-driven intelligent systems. This poses a huge challenge when it comes to deploying machine learning systems in accordance with CD principles.

MANY SOURCES OF CHANGE

+



+

Data

Schema

Sampling over Time

Volume

...

Model

Research, Experiments

Training on New Data

Performance

...

Code

New Features

Bug Fixes

Dependencies

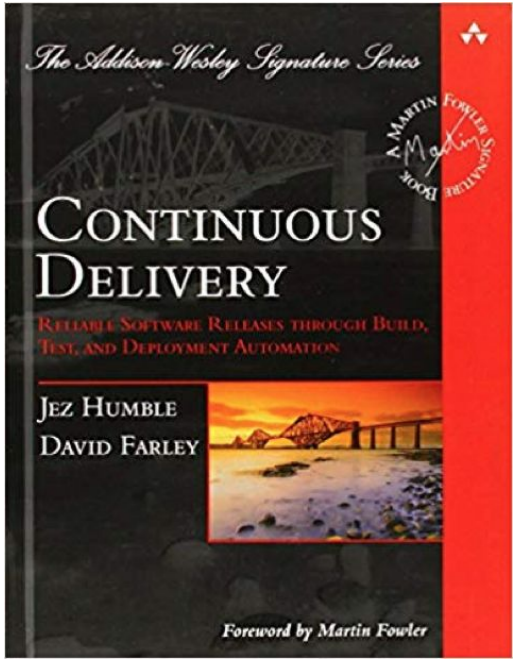
...

Icons created by Noura Mbarki and I Putu Kharismayadi from Noun Project

*"Continuous Delivery is the ability to get **changes of all types** — including new features, configuration changes, bug fixes and experiments — into production, or into the hands of users, **safely and quickly** in a **sustainable way.**"*

- Jez Humble & Dave Farley

PRINCIPLES OF CONTINUOUS DELIVERY



→ Create a Repeatable, Reliable Process for Releasing Software

→ Automate Almost Everything

→ Build Quality In

→ Work in Small Batches

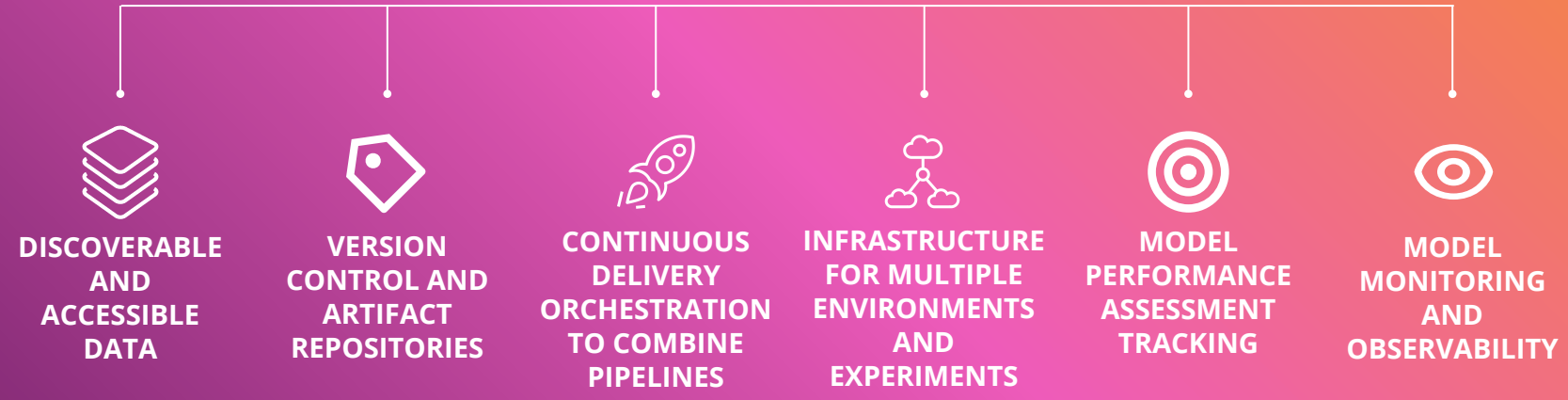
→ Keep Everything in Source Control

→ Done Means “Released”

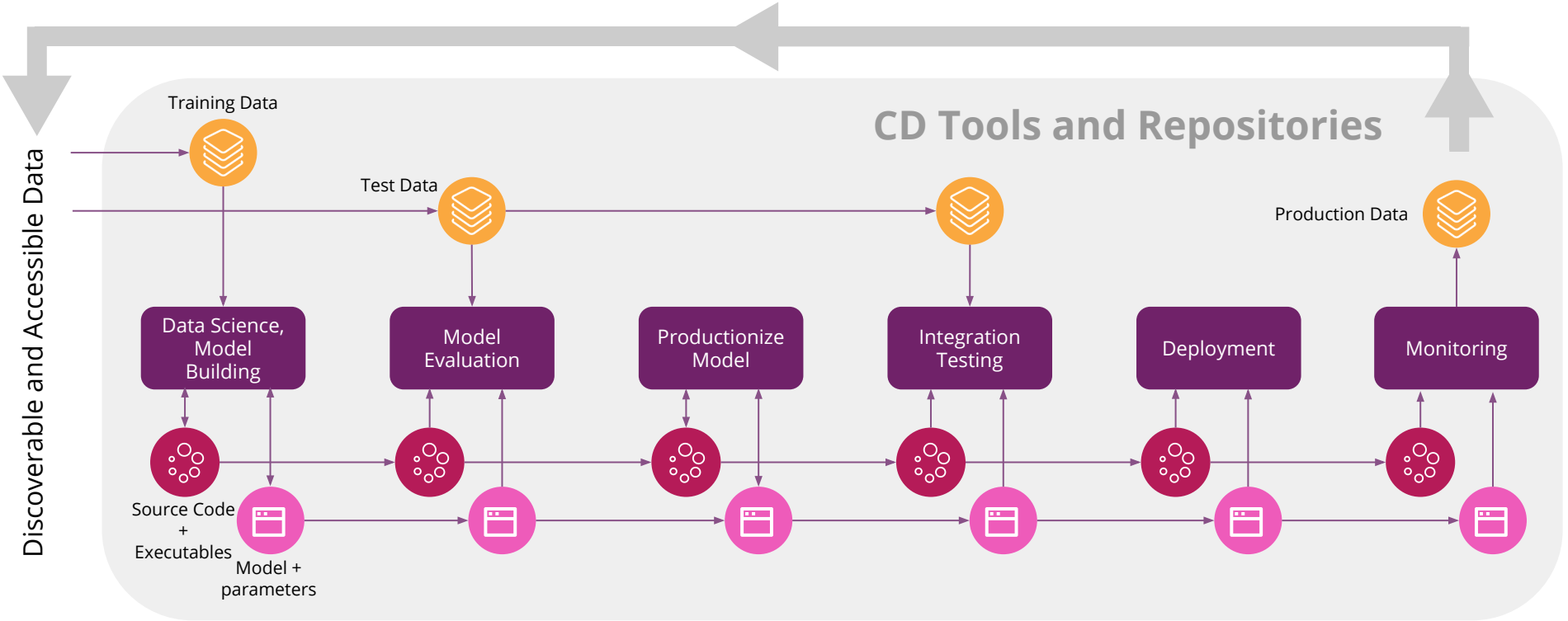
→ Improve Continuously

WHAT DO WE NEED IN OUR STACK?

Doing CD with Machine Learning is still a hard problem



PUTTING EVERYTHING TOGETHER



WHAT WE WILL USE IN THIS WORKSHOP

There are many options for tools and technologies to implement CD4ML

THE MACHINE LEARNING PROBLEM WE ARE EXPLORING TODAY

A REAL BUSINESS PROBLEM

RETAIL / SUPPLY CHAIN

Loss of sales, opportunity cost, stock waste, discounting

REQUIRES

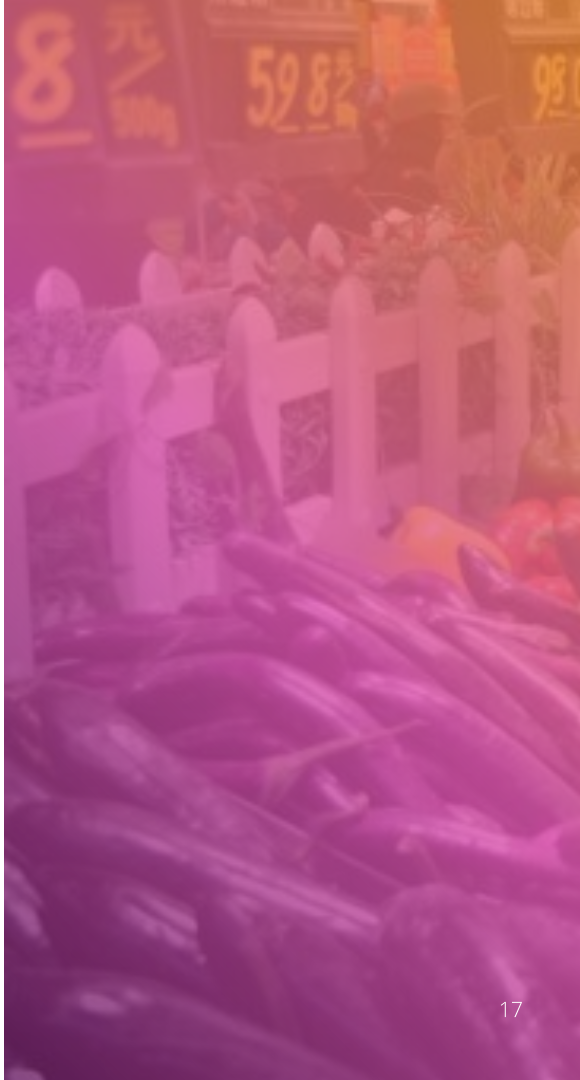
Accurate Demand Forecasting

TYPICAL CHALLENGES

→ Predictions Are Inaccurate

→ Development Takes A Long Time

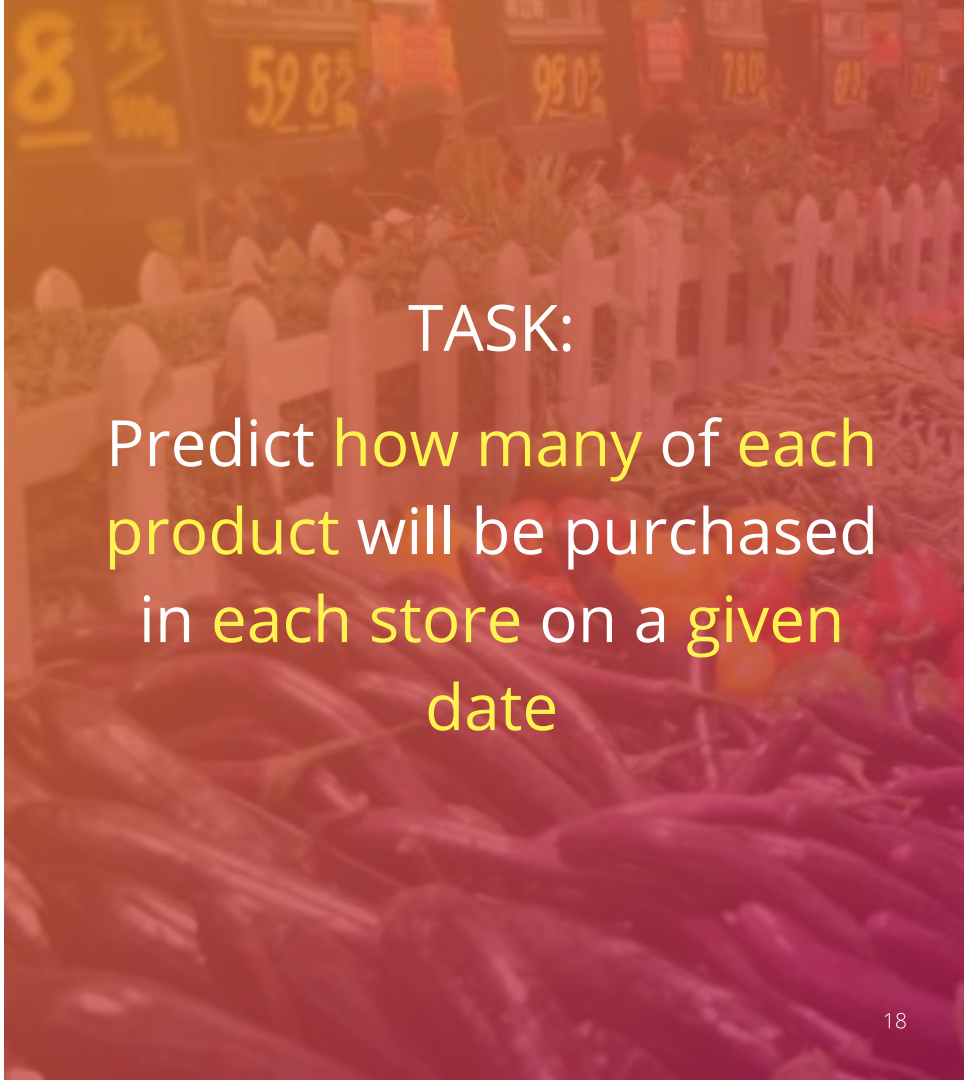
→ Difficult To Adapt To Market Change Pace



SALES FORECASTING FOR GROCERY RETAILER

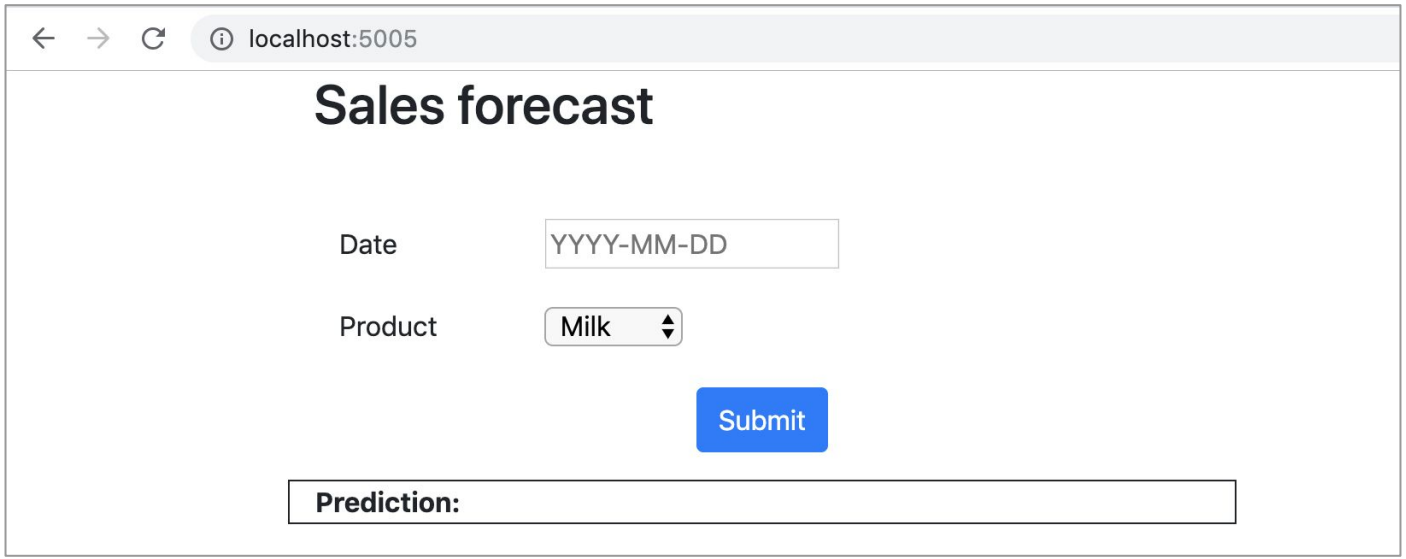
Make predictions based on data from:

- 4,000 items
- 50 stores
- 125,000,000 sales transactions
- 4.5 years of data



THE SIMPLIFIED WEB APPLICATION

As a buyer, I want to be able to choose a product and predict how many units the product will sell at a future date.



EXERCISE 1: SETUP

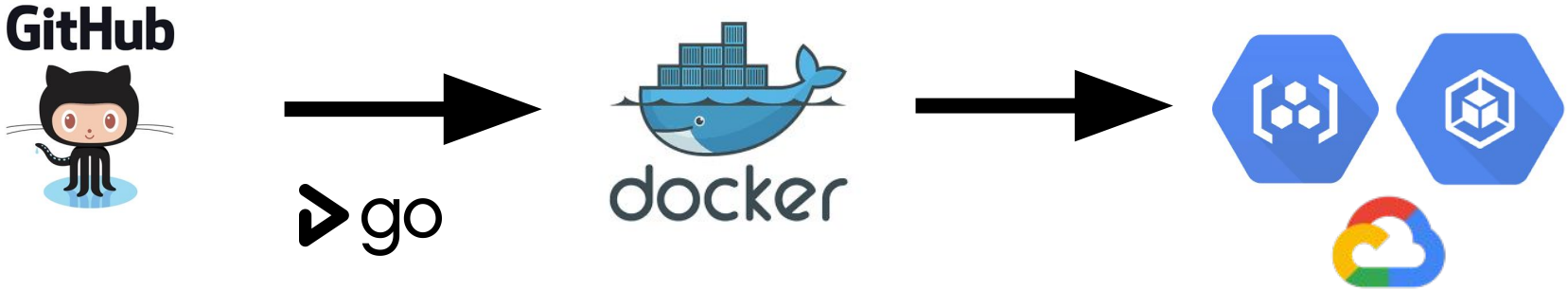
https://github.com/ThoughtWorksInc/continuous-intelligence-workshop

- Click on `instructions → 1-setup.md`
- Follow the steps to setup your local development environment

User ID assignment sheet: http://bit.ly/cd4ml-strata19

DEPLOYMENT PIPELINE

Automates the process of building, testing, and deploying applications to production

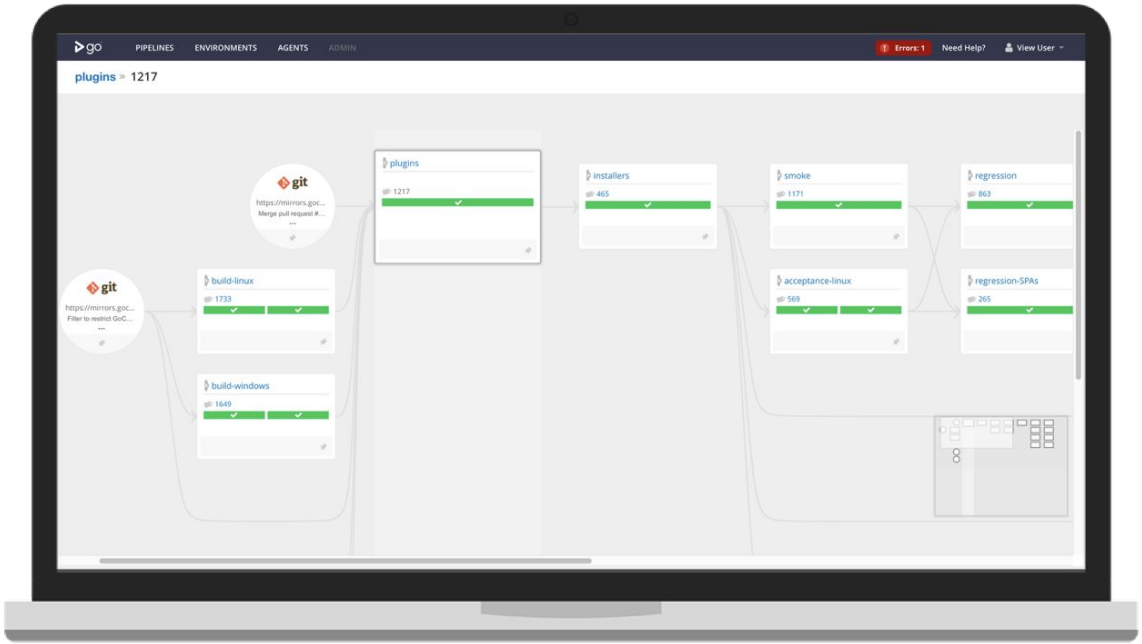


Application **code** in version control repository

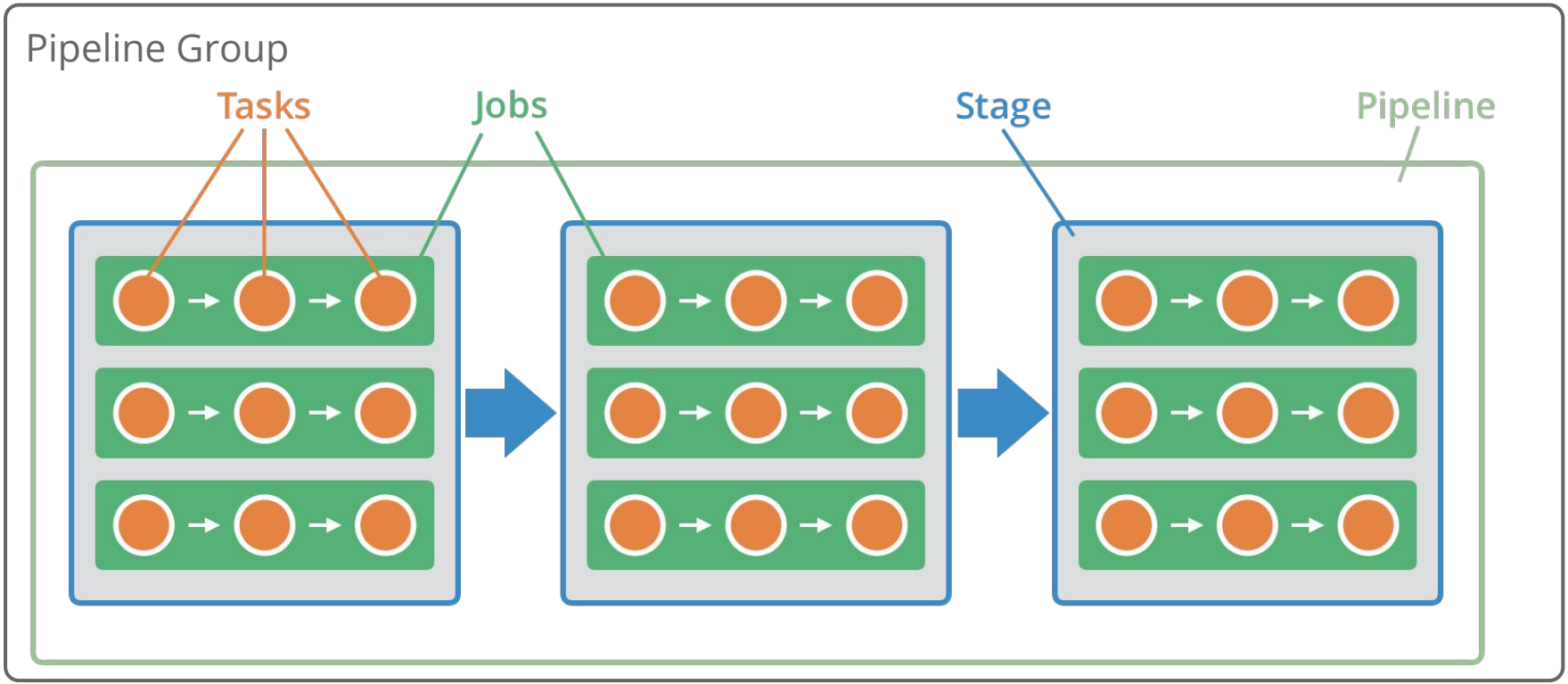
Container image as **deployment artifact**

Deploy container to **production servers**

An Open Source Continuous Delivery server to model and visualise complex workflows



ANATOMY OF A GOCD PIPELINE



EXERCISE 2: DEPLOYMENT PIPELINE

https://github.com/ThoughtWorksInc/continuous-intelligence-workshop

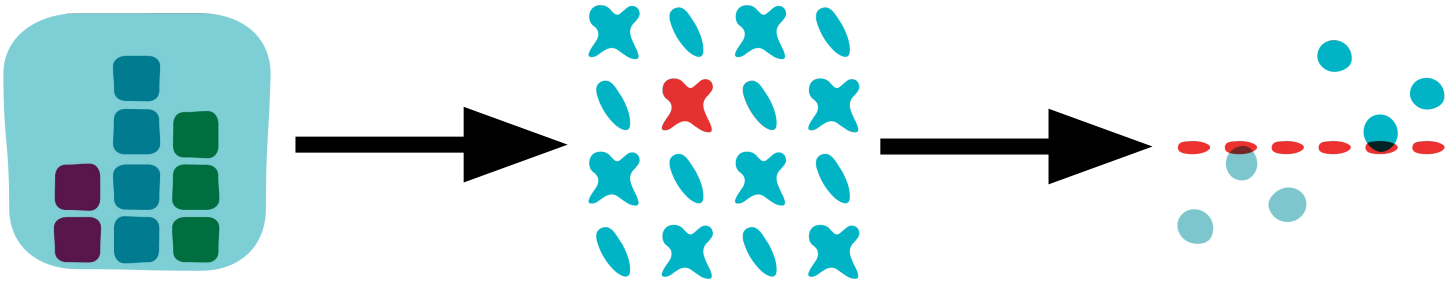
- Click on `instructions → 2-deployment-pipeline.md`
- Follow the steps to setup your deployment pipeline
- GoCD URL: https://gocd.cd4ml.net

BUT NOW WHAT?

Once your model is in production…

- How do we retrain the model more often?
- How to deploy the retrained model to production?
- How to make sure we don't break anything when deploying?
- How to make sure that our modeling approach or parameterization is still the best fit for the data?
- How to monitor our model "in the wild"?

BASIC DATA SCIENCE WORKFLOW

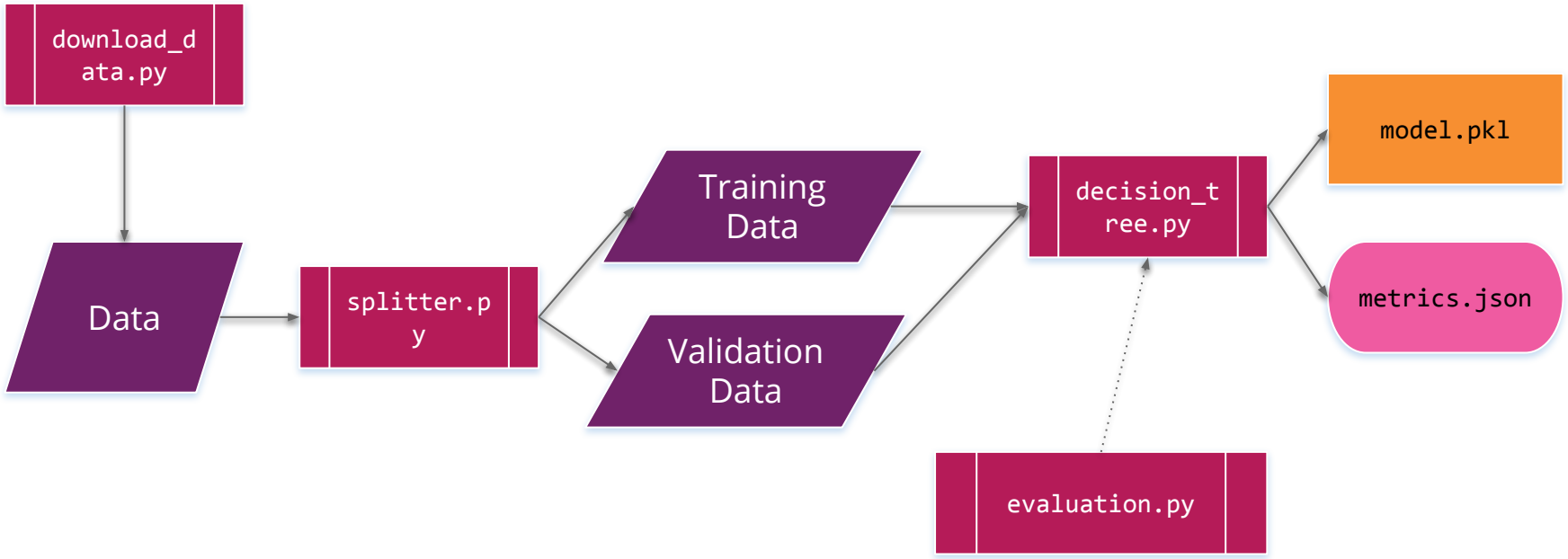


Gather data and extract features

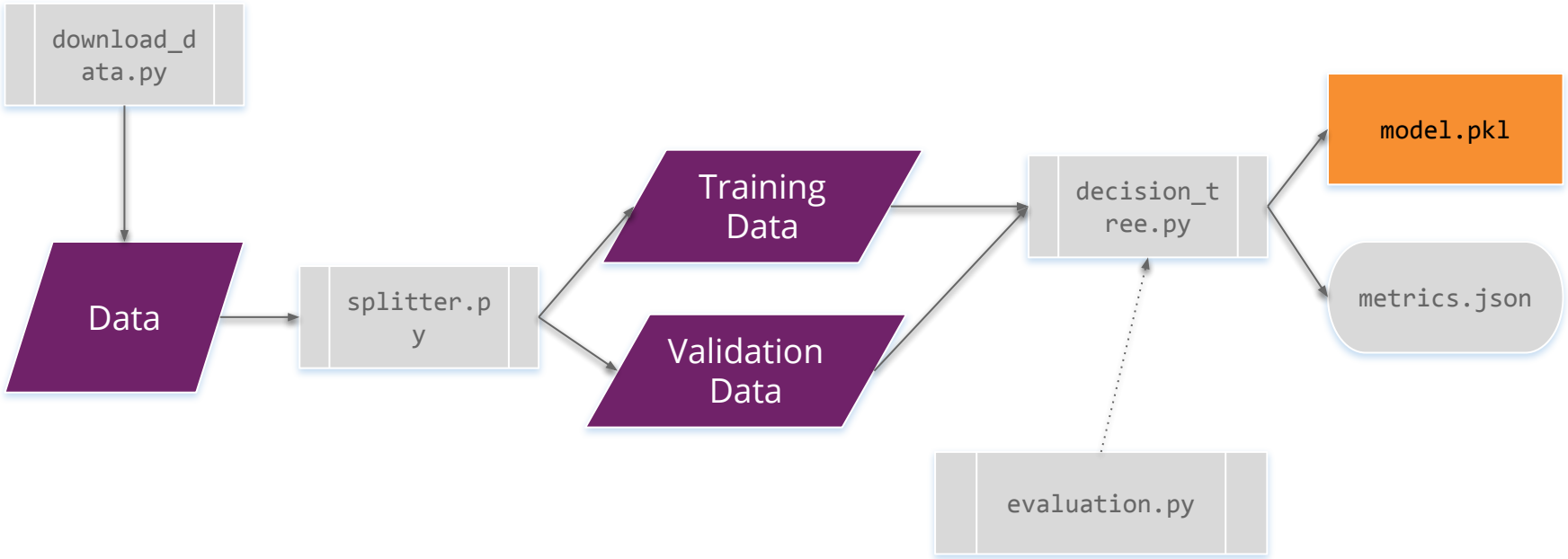
Separate into training and validation sets

Train model and evaluate performance

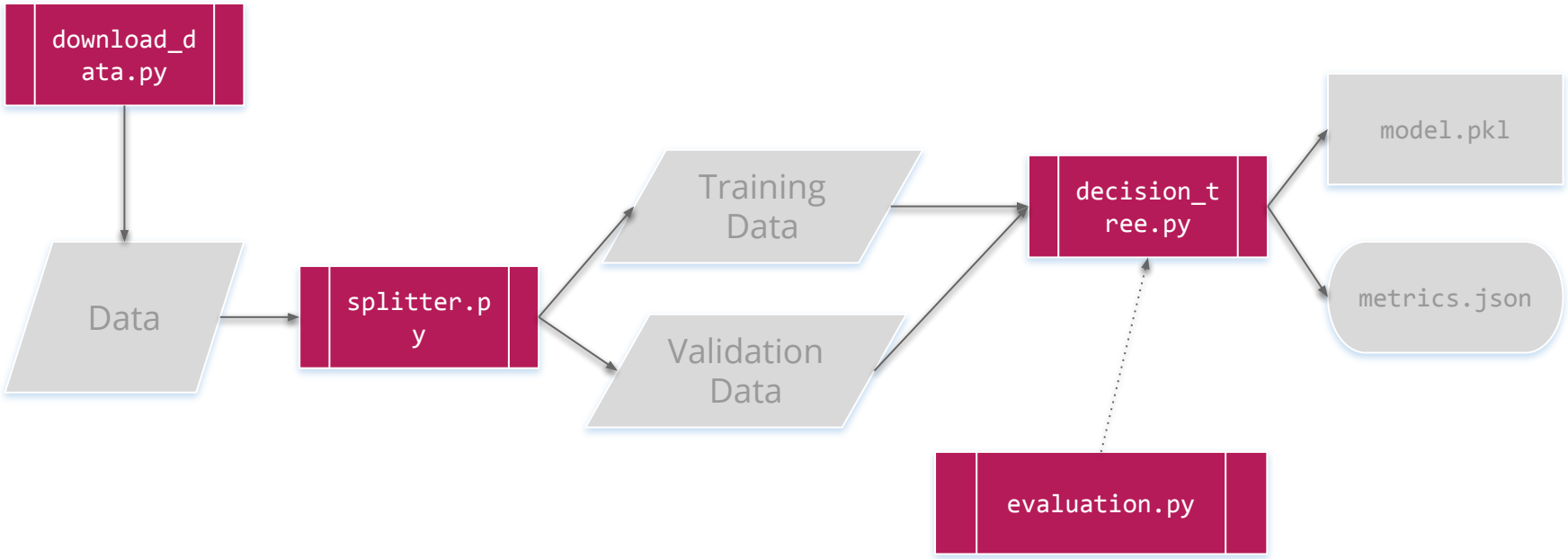
SALES FORECAST MODEL TRAINING PROCESS



CHALLENGE 1: THESE ARE LARGE FILES!



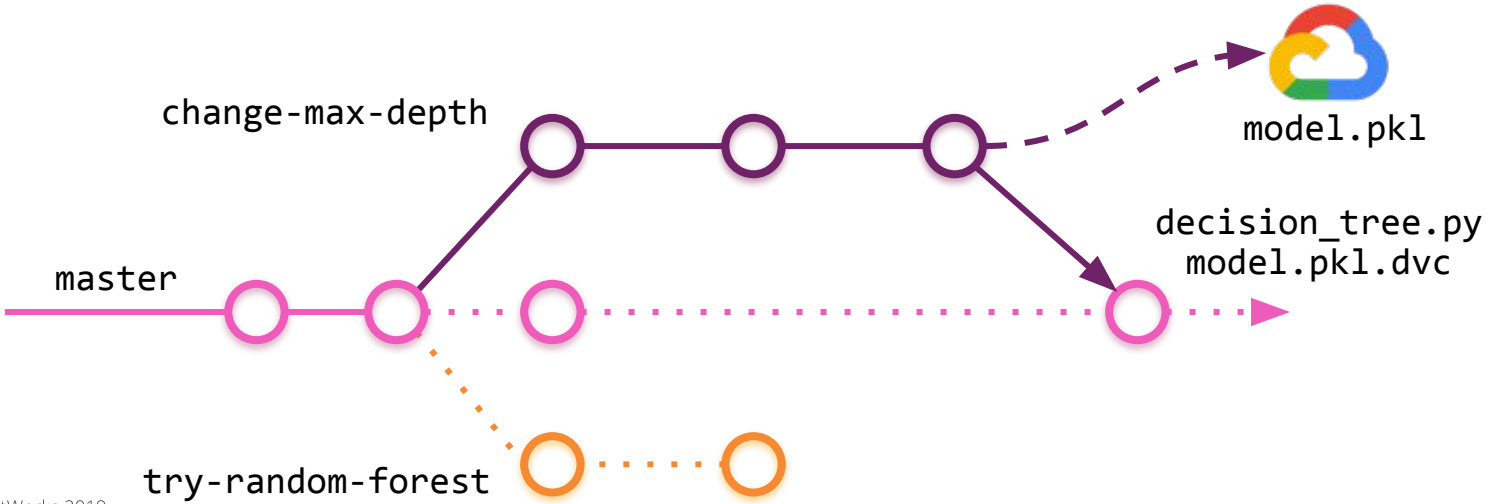
CHALLENGE 2: AD-HOC MULTI-STEP PROCESS



SOLUTION: dvc

***d**ata science **v**ersion **c**ontrol*

- dvc is git porcelain for storing large files using cloud storage
- dvc connects model training steps to create reproducible workflows



ANATOMY OF A DVC COMMAND

```
dvc run -d src/download_data.py
-o data/raw/store47-2016.csv python src/download_data.py
```

This runs a **command** and creates a `.dvc` file. The dvc file points to the **dependencies**. The **output files** are versioned and stored in the cloud by running `dvc push`.

When you use the output files (`store47-2016.csv`) as dependencies for the next step, a *pipeline* is automatically created.

You can re-execute an entire pipeline with one command: `dvc repro`

EXERCISE 3: MACHINE LEARNING PIPELINE

https://github.com/ThoughtWorksInc/continuous-intelligence-workshop

- Click on `instructions → 3-machine-learning-pipeline.md`
- Follow the steps on your local development environment and in GoCD to create your Machine Learning pipeline

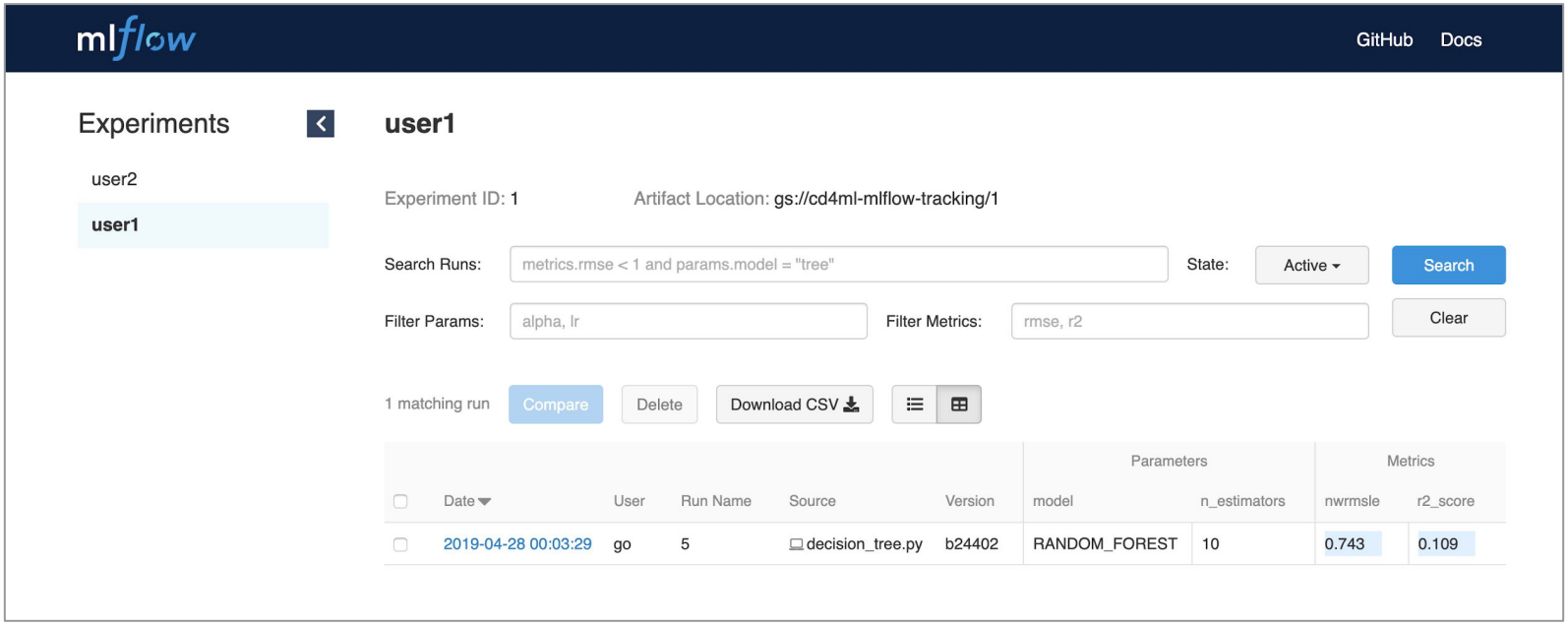
HOW DO WE TRACK EXPERIMENTS?

We need to track the scientific process and evaluate our models:

- Which experiments and hypothesis are being explored?
- Which algorithms are being used in each experiment?
- Which version of the code was used?
- How long does it take to run each experiment?
- What parameter and hyperparameters were used?
- How fast are my models learning?
- How do we compare results from different runs?

mlflow

An Open Source platform for managing end-to-end machine learning lifecycle



EXERCISE 4: TRACKING EXPERIMENTS

https://github.com/ThoughtWorksInc/continuous-intelligence-workshop

- Click on `instructions → 4-tracking-experiments.md`
- Follow the steps to track ML training in mlflow
- MLflow URL: https://mlflow.cd4ml.net

HOW TO LEARN CONTINUOUSLY?

We need to capture production data to improve our models:

- Track model usage
- Track model inputs to find training-serving skew
- Track model outputs
- Track model interpretability outputs to identify potential bias or overfit
- Track model fairness to understand how it behaves against dimensions that could introduce unfair bias

EFK STACK

Monitoring and Observability infrastructure

fluentd

Open Source **data collector** for unified logging

elasticsearch

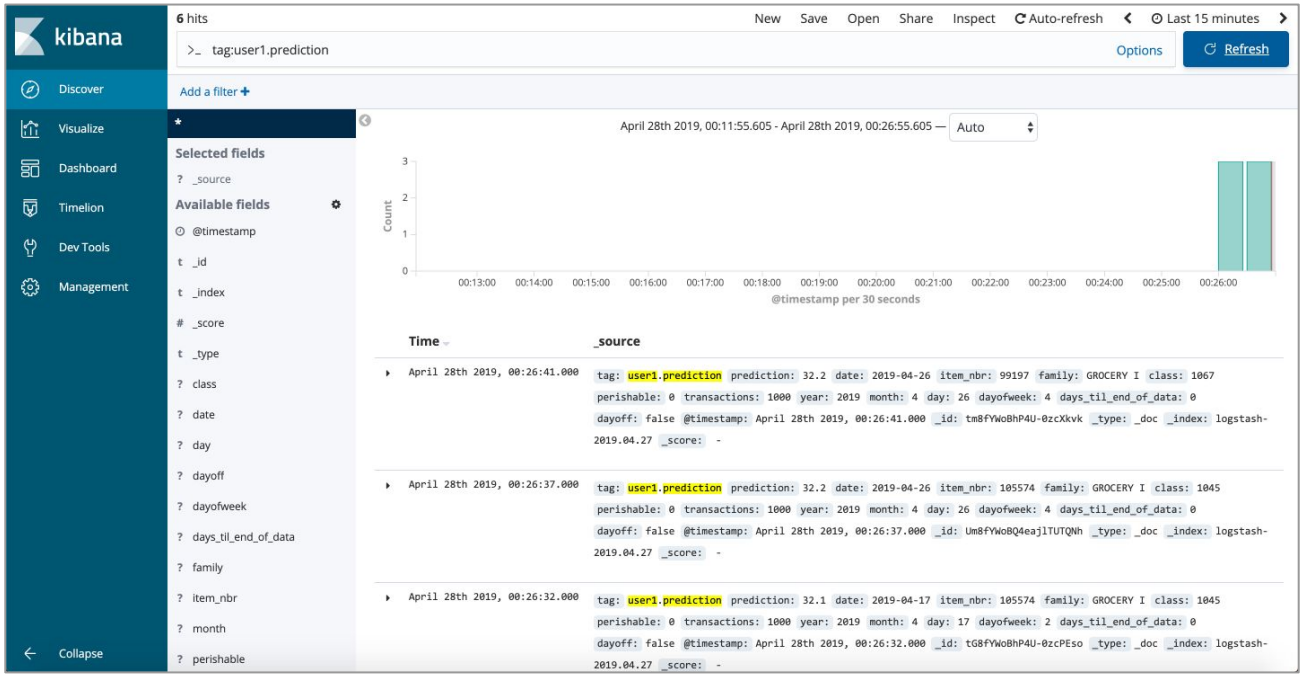
Open Source **Search Engine**

kibana

Open Source web UI to **explore and visualise data**

kibana

An Open Source UI that makes it easy to explore and visualise the data index in Elasticsearch



EXERCISE 5: MODEL MONITORING

https://github.com/ThoughtWorksInc/continuous-intelligence-workshop

- Click on `instructions → 5-model-monitoring.md`
- Follow the steps to log prediction events
- Kibana URL: https://kibana.cd4ml.net

SUMMARY - WHAT HAVE WE LEARNED?

CD4ML

- Proper data/model versioning tools enable reproducible work to be done in parallel.
- No need to maintain complex data processing/model training scripts.
- We can then put data science work into a Continuous Delivery workflow.
- **Result: Continuous, on-demand AI development and deployment, from research to production, with a single command.**
- **Benefit: production AI systems that are always as smart as your data science team.**

join.thoughtworks.com

THANK YOU!

Danilo Sato (dsato@thoughtworks.com)
Christoph Windheuser (cwindheu@thoughtworks.com)
Emily Gorcenski (egorcens@thoughtsworks.com)
Arif Wider (awider@thoughtworks.com)

ThoughtWorks®