RED HAT SUMMIT

MACHINE LEARNING WORKFLOWS FOR APPLICATION DEVELOPERS

Sophie Watson • @sophwats • sophie@redhat.com
William Benton • @willb • willb@redhat.com
Michael McCune • @FOSSJunkie • elmiko@redhat.com

What you’ll learn today

Processes and principles to solve problems with ML

Some tricks to visualize complex data

How to train, evaluate, and deploy ML models

How OpenShift makes it all easier

Legal disclaimer

The content set forth herein does not constitute in any way a binding or legal agreement or impose any legal obligation or duty on Red Hat. This information is provided for discussion purposes only and is subject to change for any or no reason.

Forecast

What is machine learning?

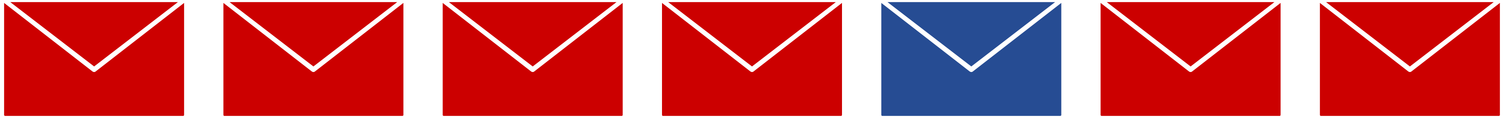
What workflow do machine learning practitioners use?

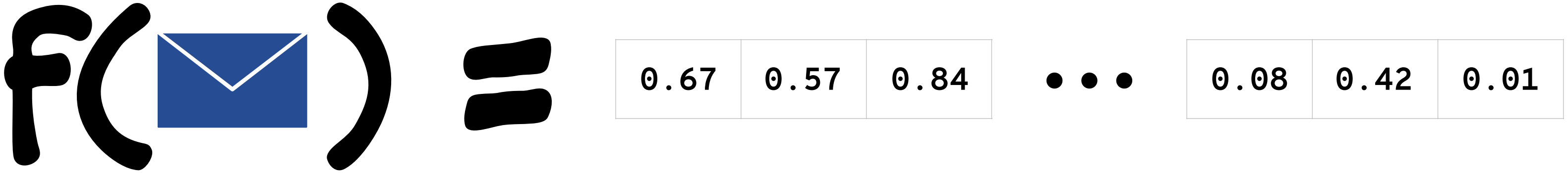
How can we incorporate machine learning into apps?

What do we have to look forward to?

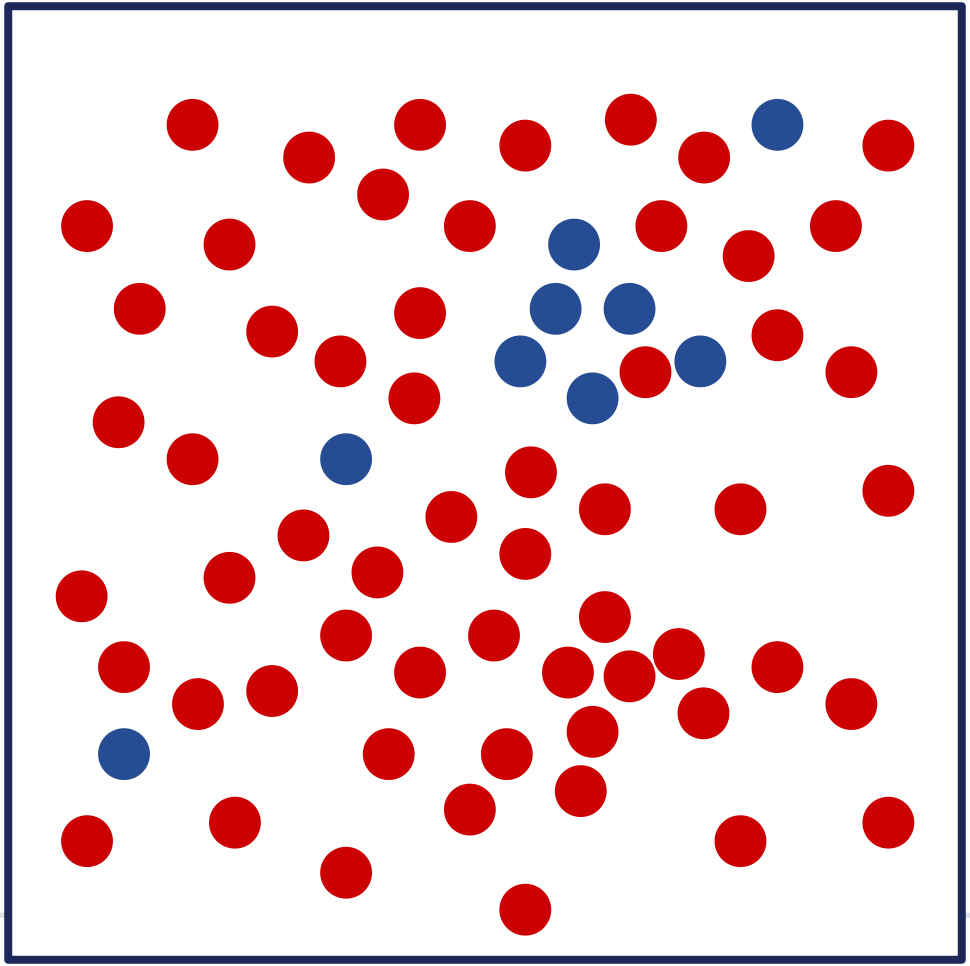
What is machine learning?

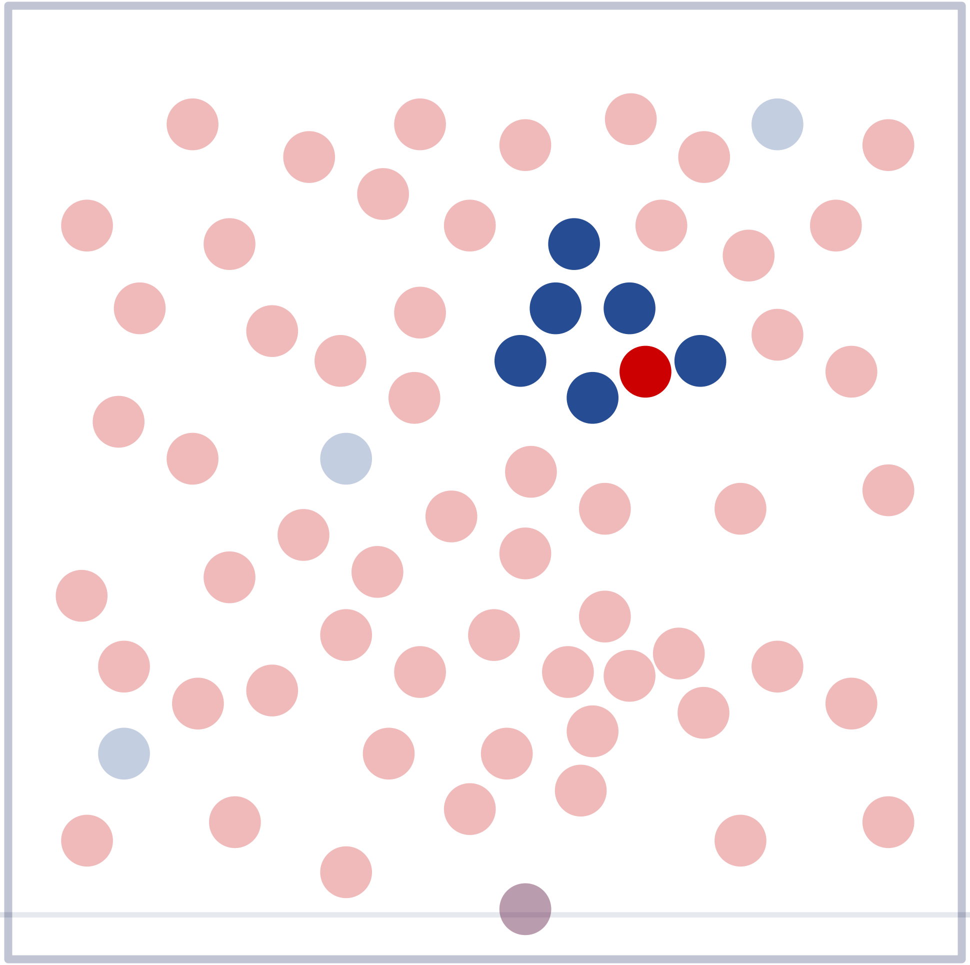
Machine learning is a family of techniques to automatically derive executable functions from example inputs and outputs.

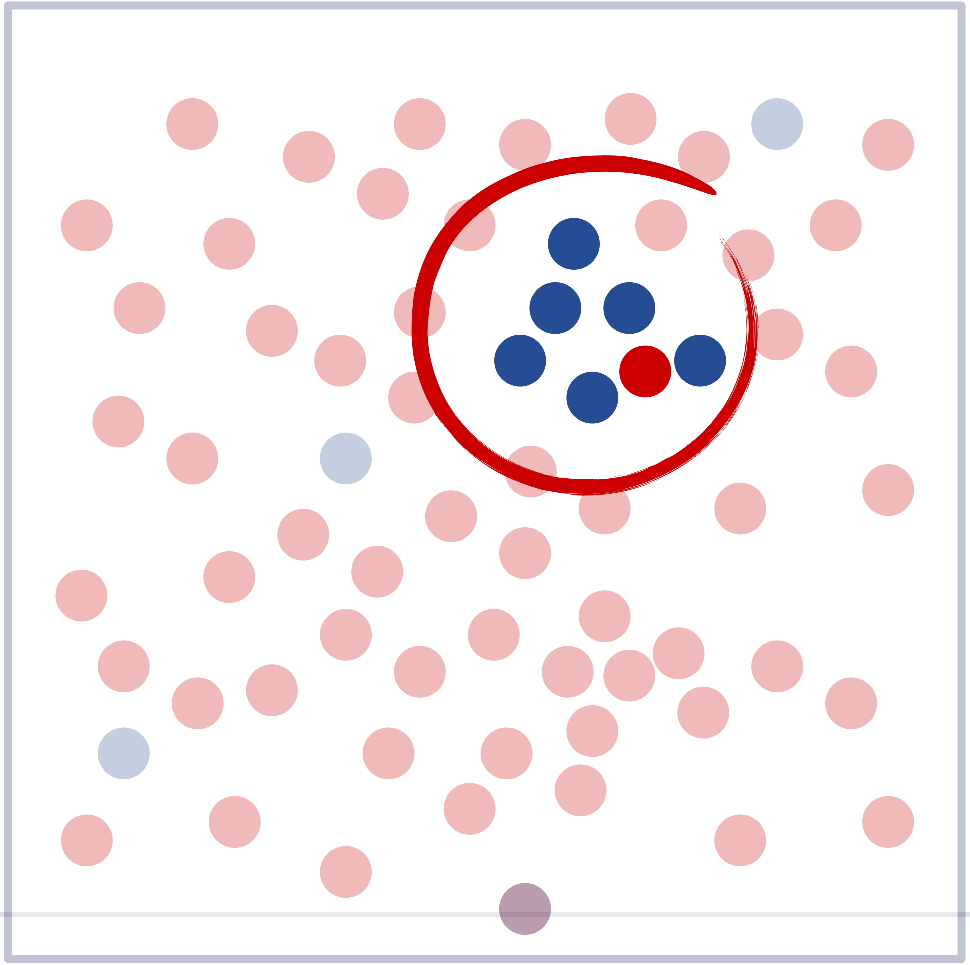


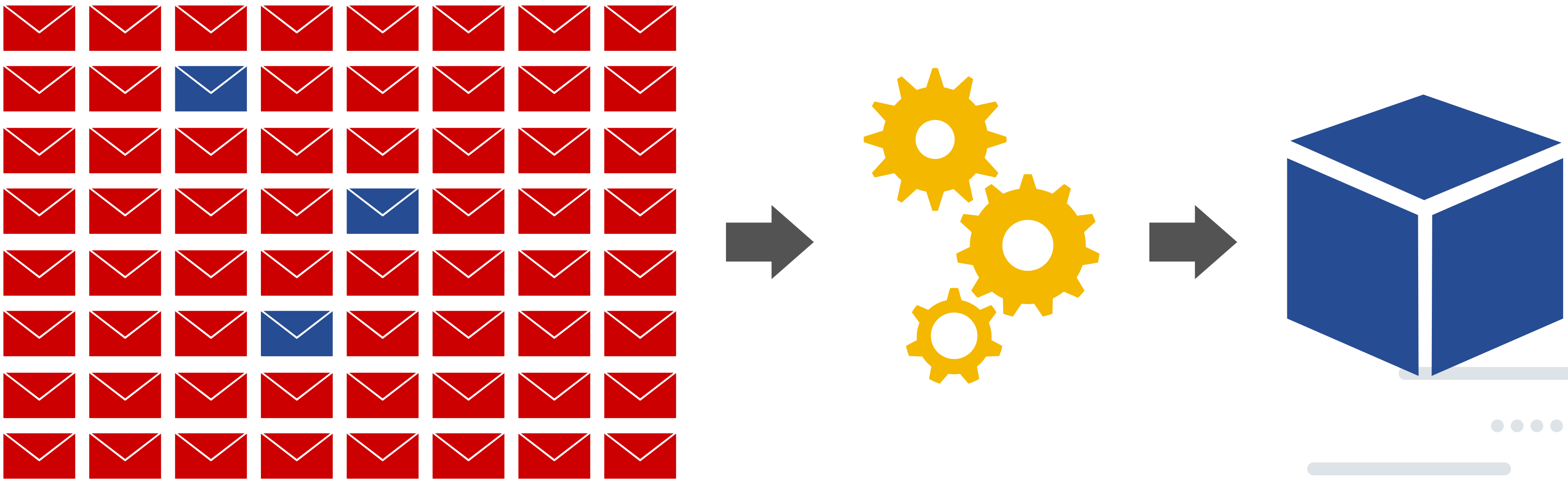


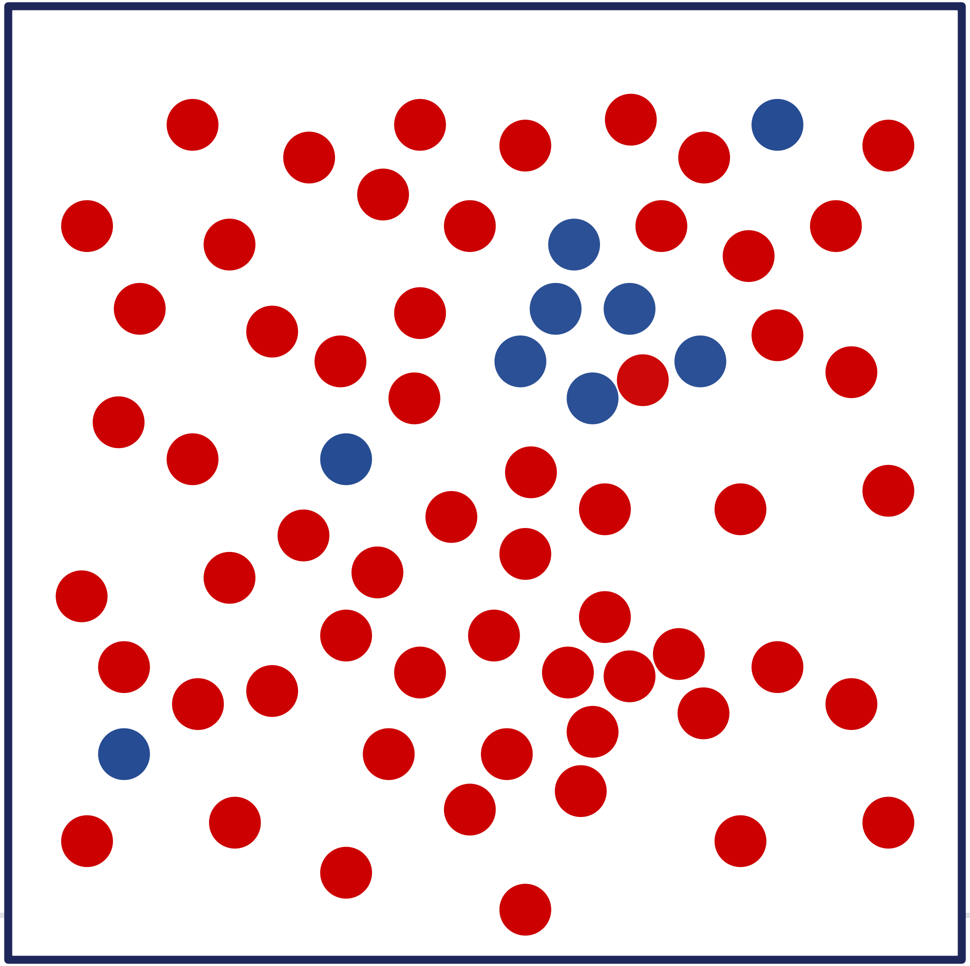
0.67	0.57	0.84	• • •	0.08	0.42	0.01

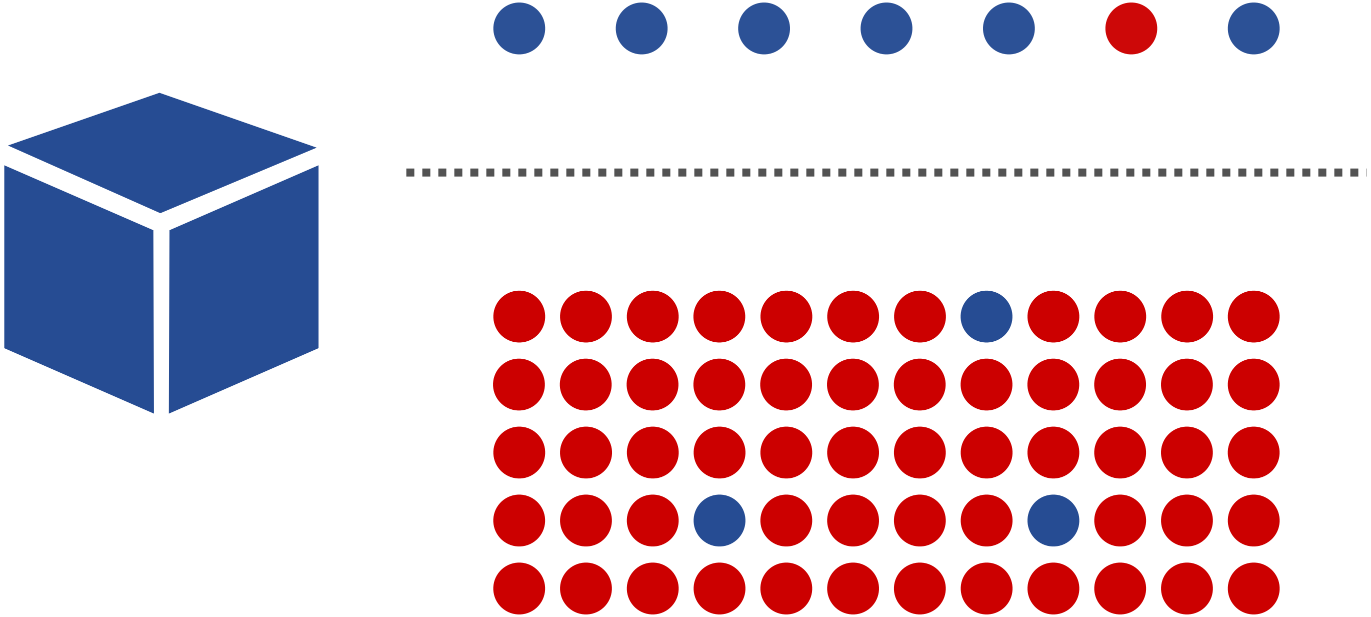












Machine learning workflows

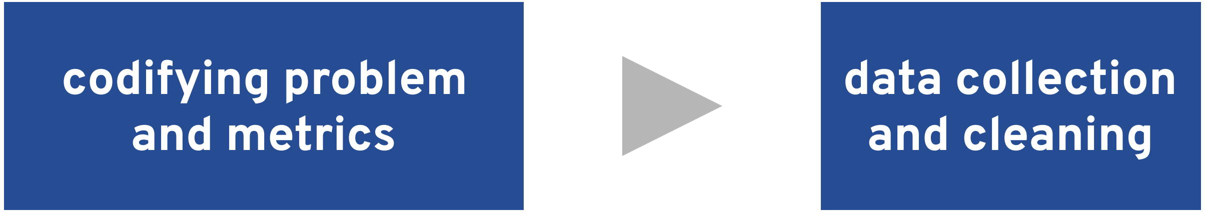
Machine learning checklist

Example data

A training objective and a business metric

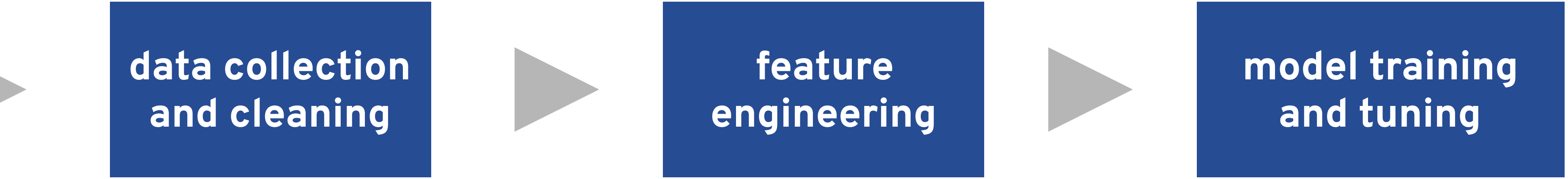
A concrete learning technique

A way to encode data



codifying problem and metrics

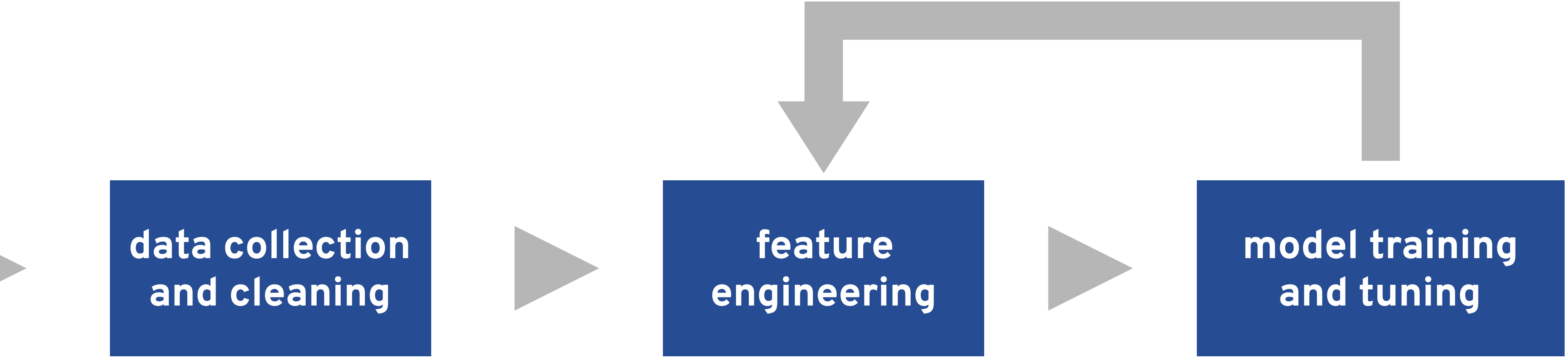
data collection and cleaning



data collection and cleaning

feature engineering

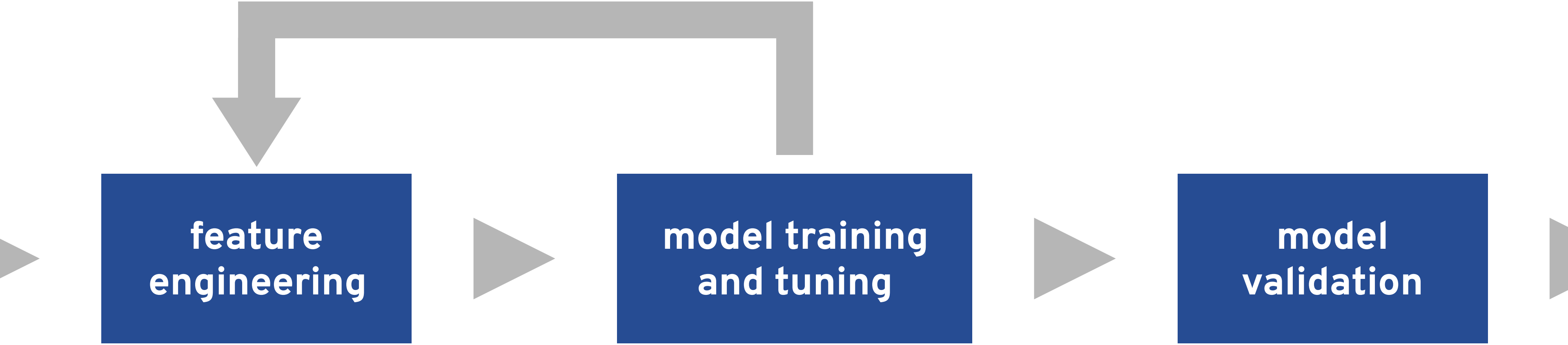
model training and tuning



data collection and cleaning

feature engineering

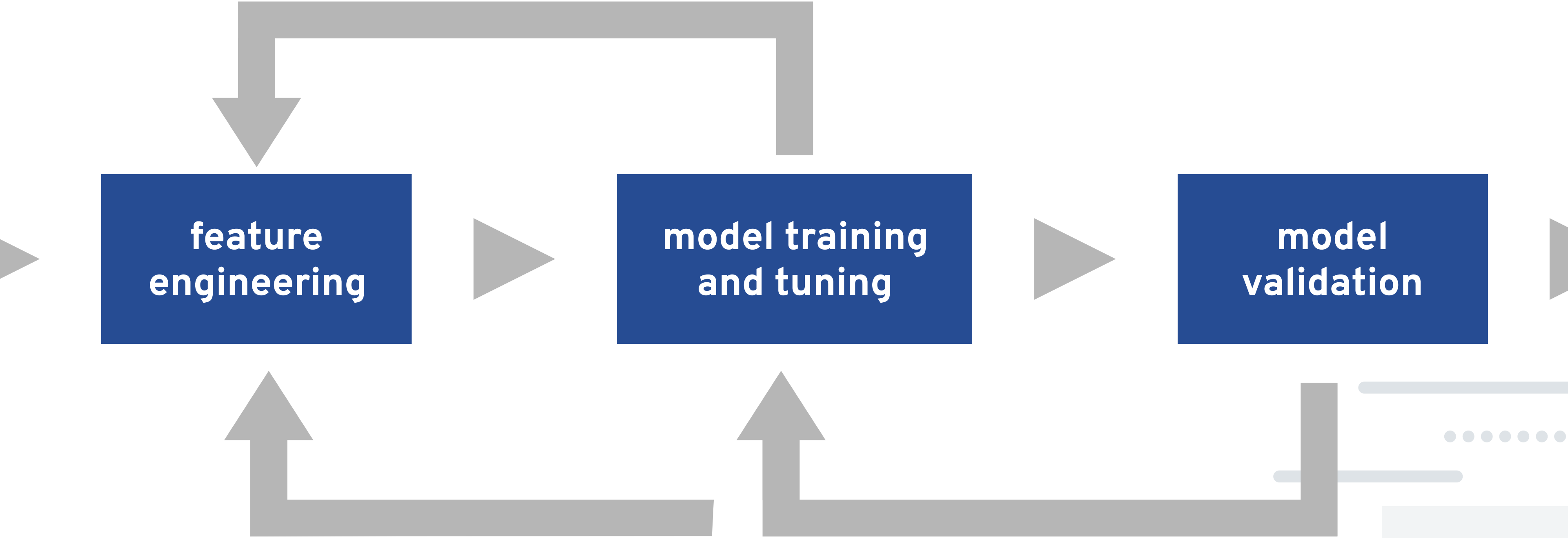
model training and tuning



feature engineering

model training and tuning

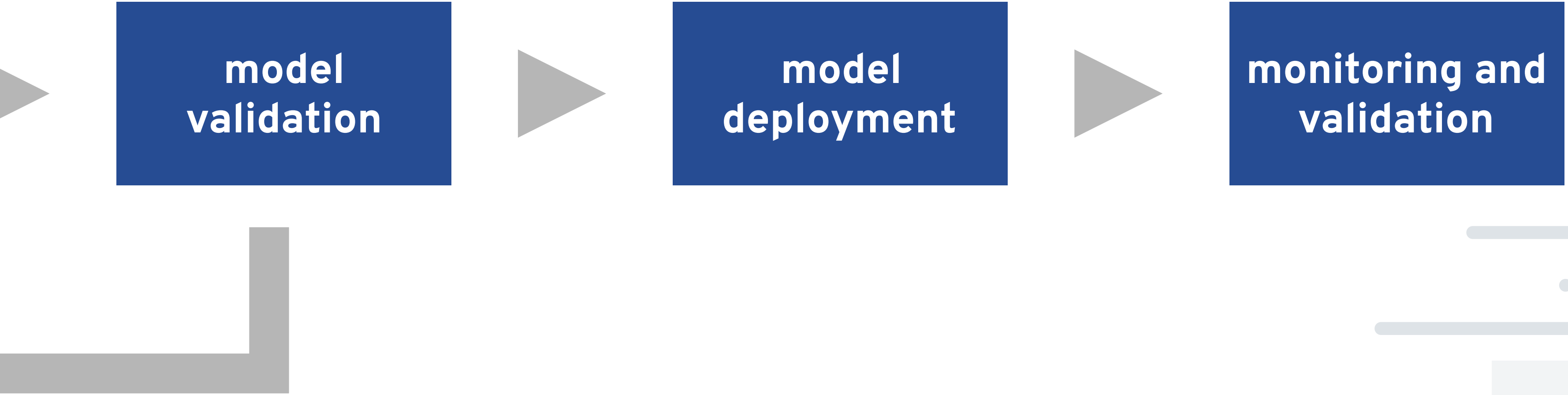
model validation



feature engineering

model training and tuning

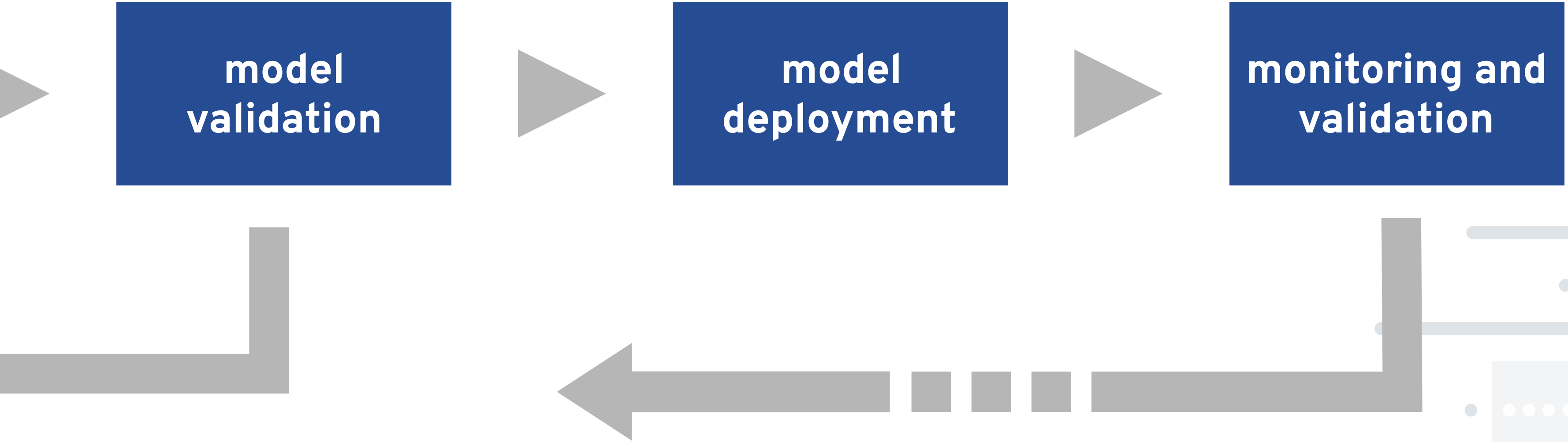
model validation

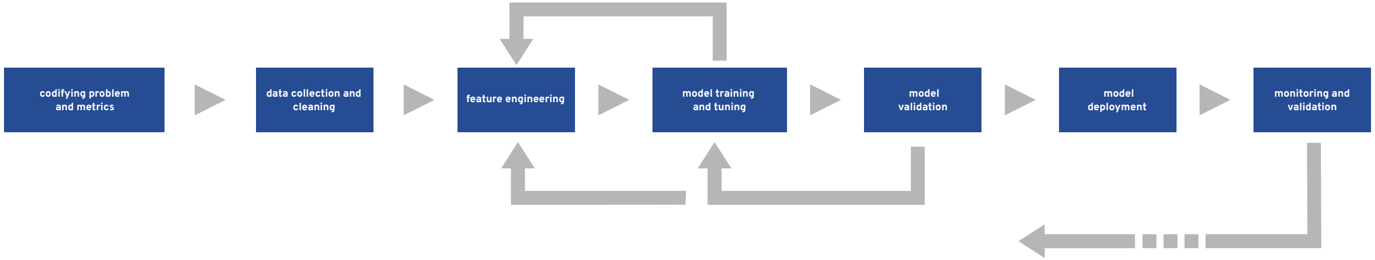


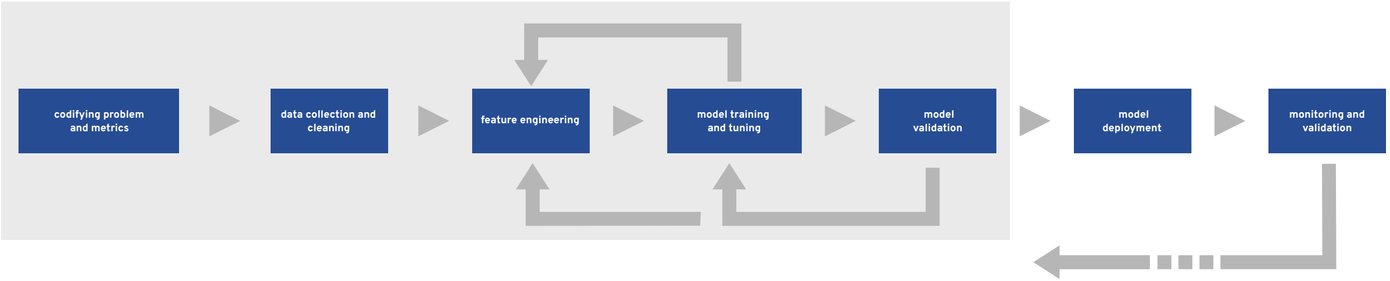
model validation

model deployment

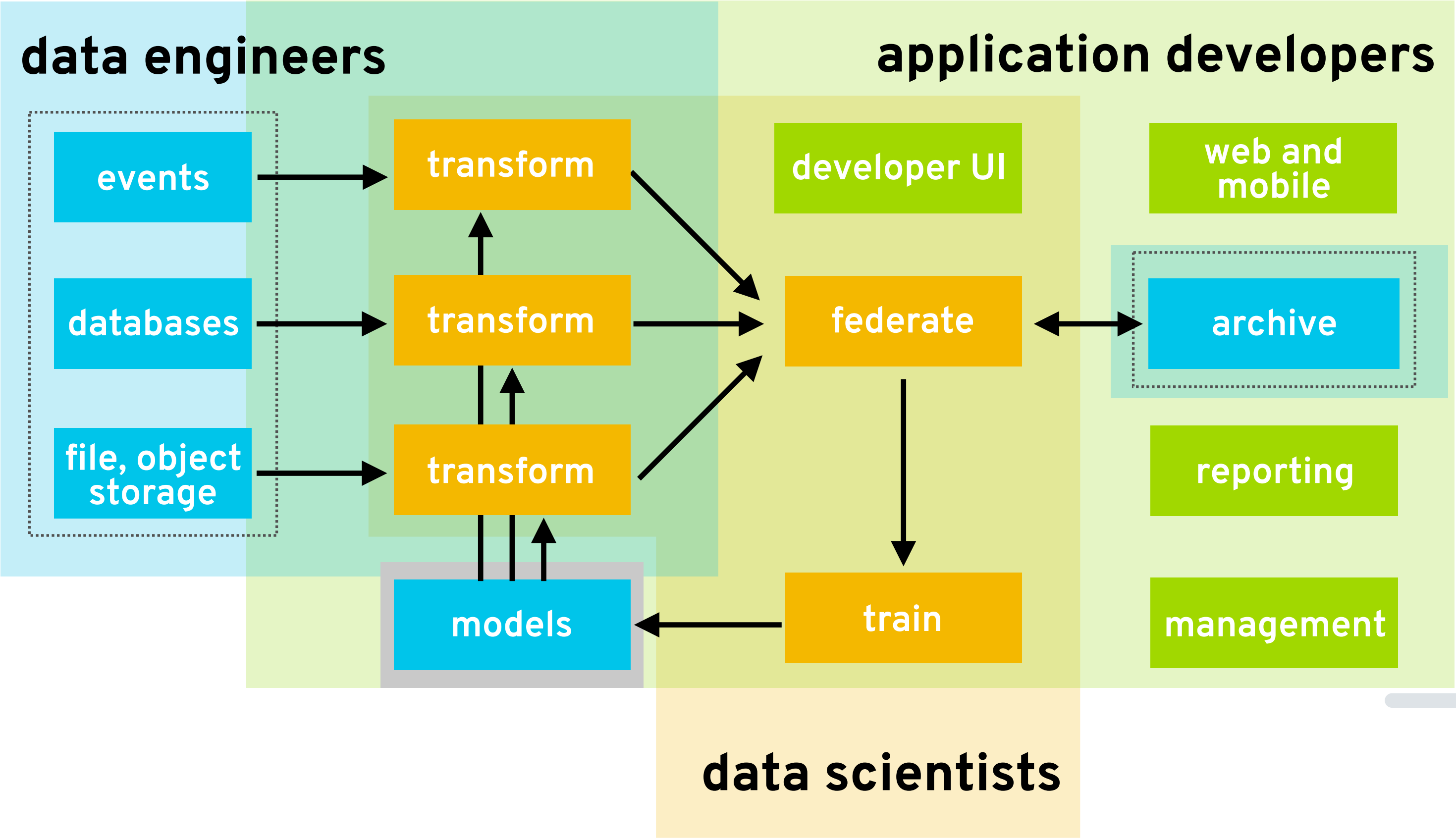
monitoring and validation


model validation
model deployment
monitoring and validation

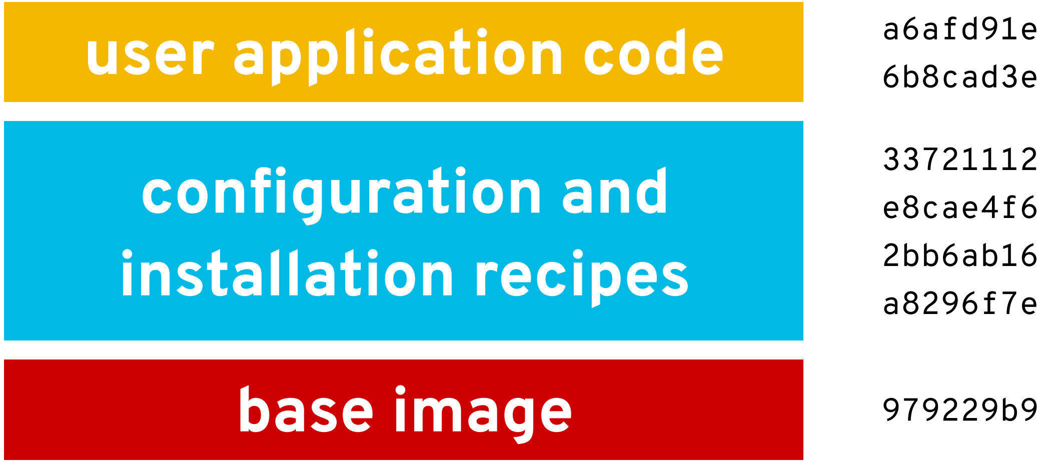

codifying problem and metrics
data collection and cleaning
feature engineering
model training and tuning
model validation
model deployment
monitoring and validation


codifying problem and metrics
data collection and cleaning
feature engineering
model training and tuning
model validation
model deployment
monitoring and validation

Machine learning, apps, and OpenShift


data engineers
application developers
events
databases
file, object storage
transform
transform
transform
developer UI
federate
train
models
web and mobile
archive
reporting
management
data scientists

Immutable images



user application code

configuration and installation recipes

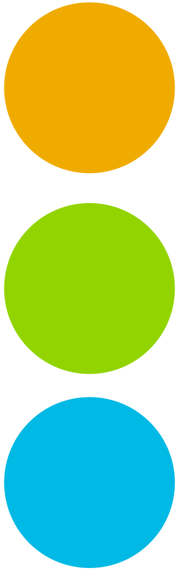
base image

a6afd91e
6b8cad3e

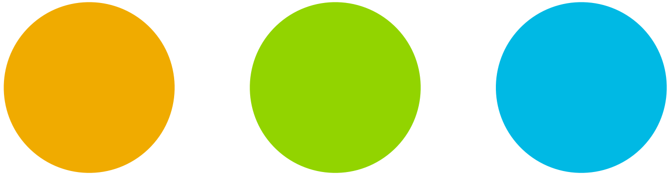
33721112
e8cae4f6
2bb6ab16
a8296f7e

979229b9

Stateless microservices

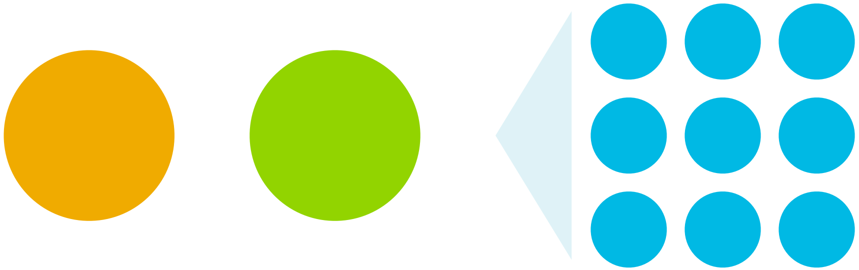


Stateless microservices

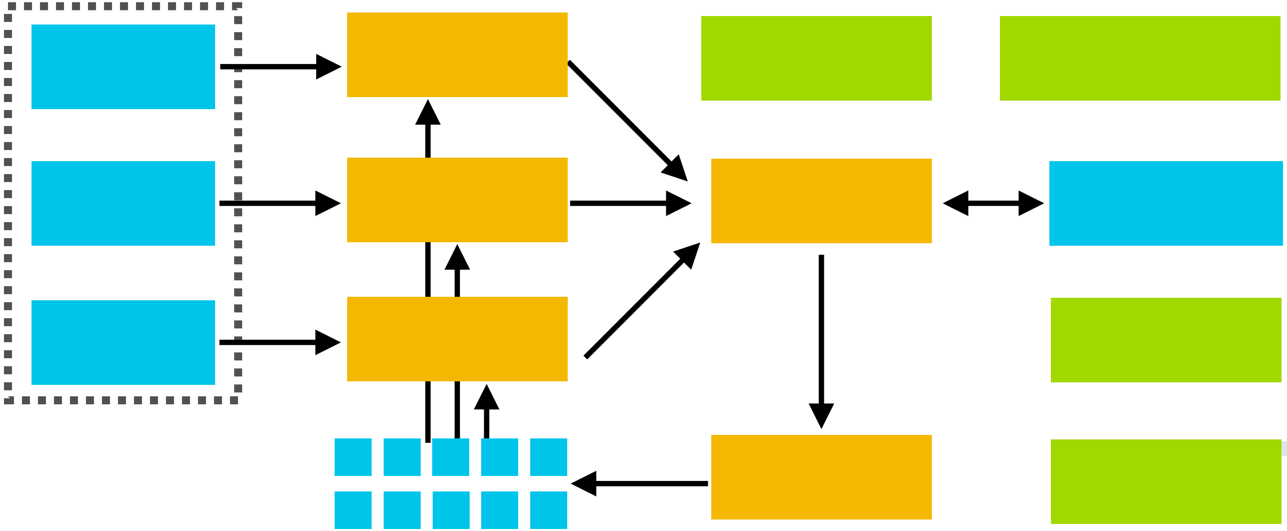


Stateless microservices

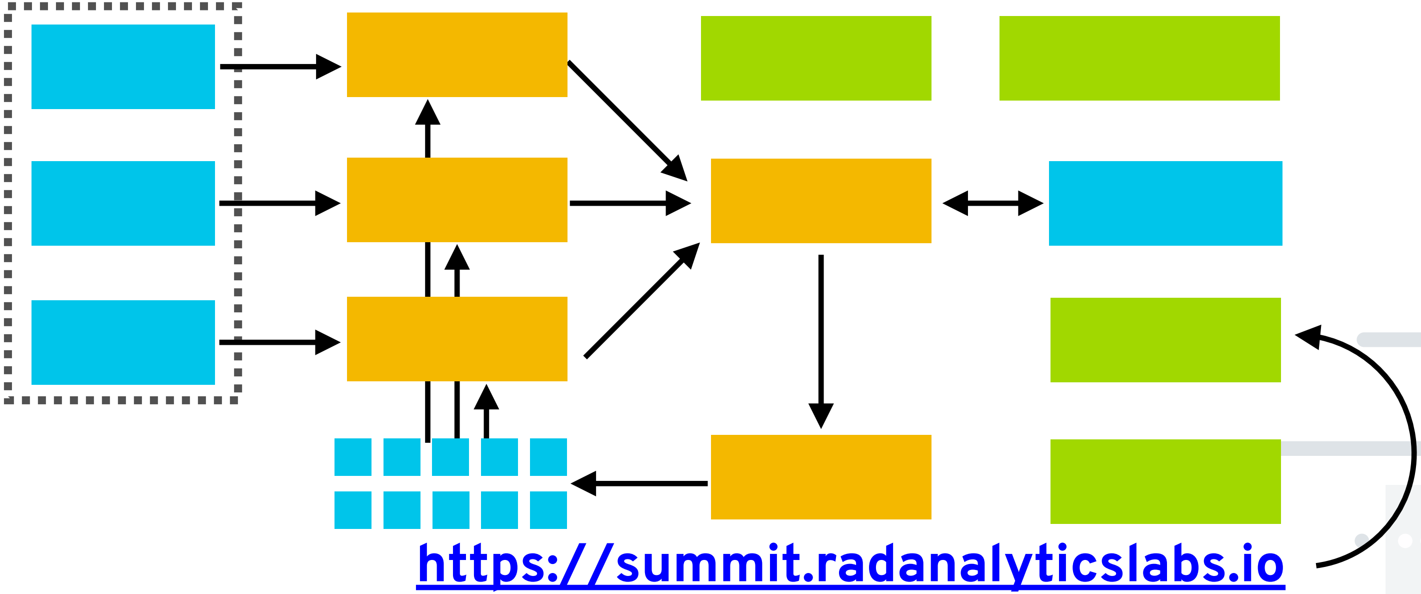
Stateless microservices



Declarative app configuration



Declarative app configuration

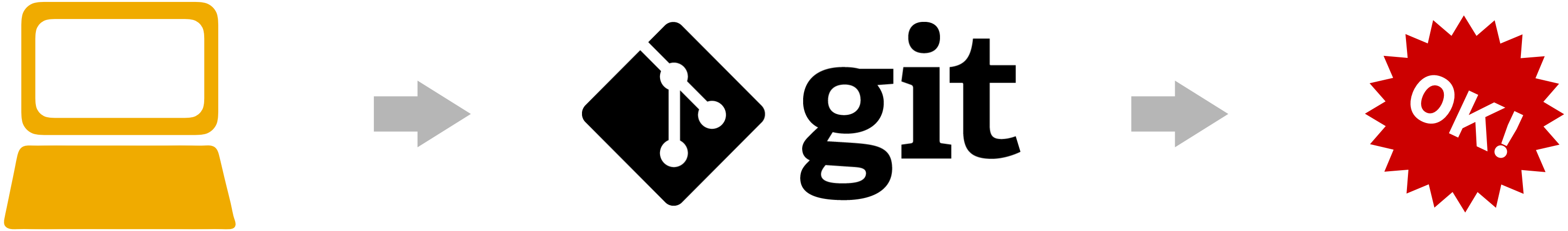


https://summit.radanalyticslabs.io

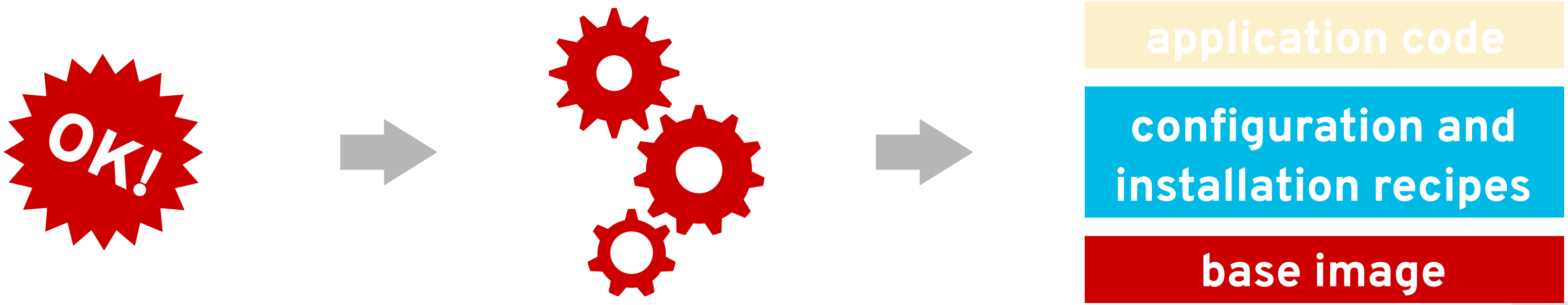
#redhat #rhsummit

Integration and deployment

Integration and deployment



Integration and deployment

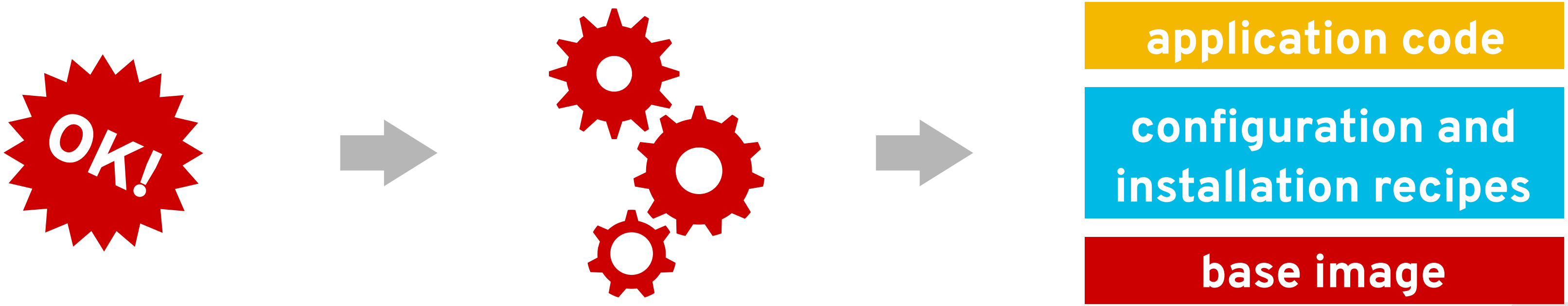


application code

configuration and installation recipes

base image

Integration and deployment

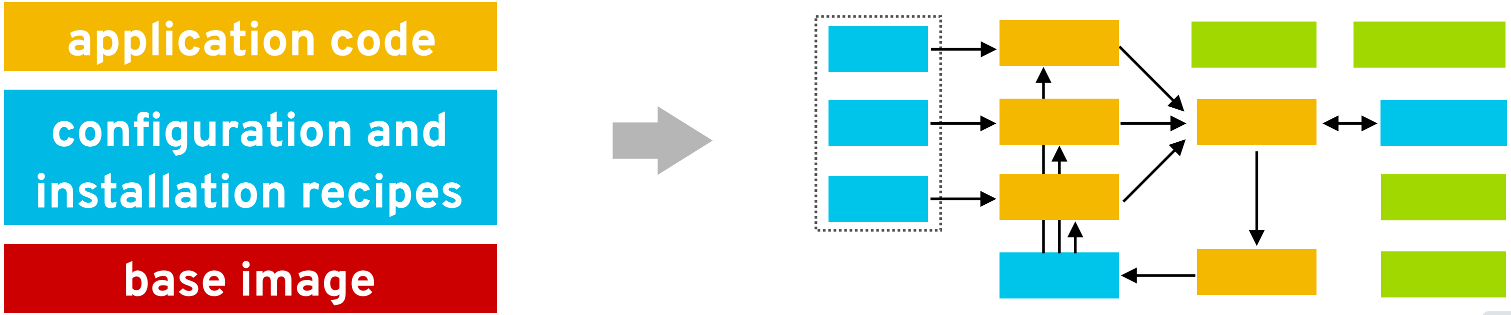


application code

configuration and installation recipes

base image

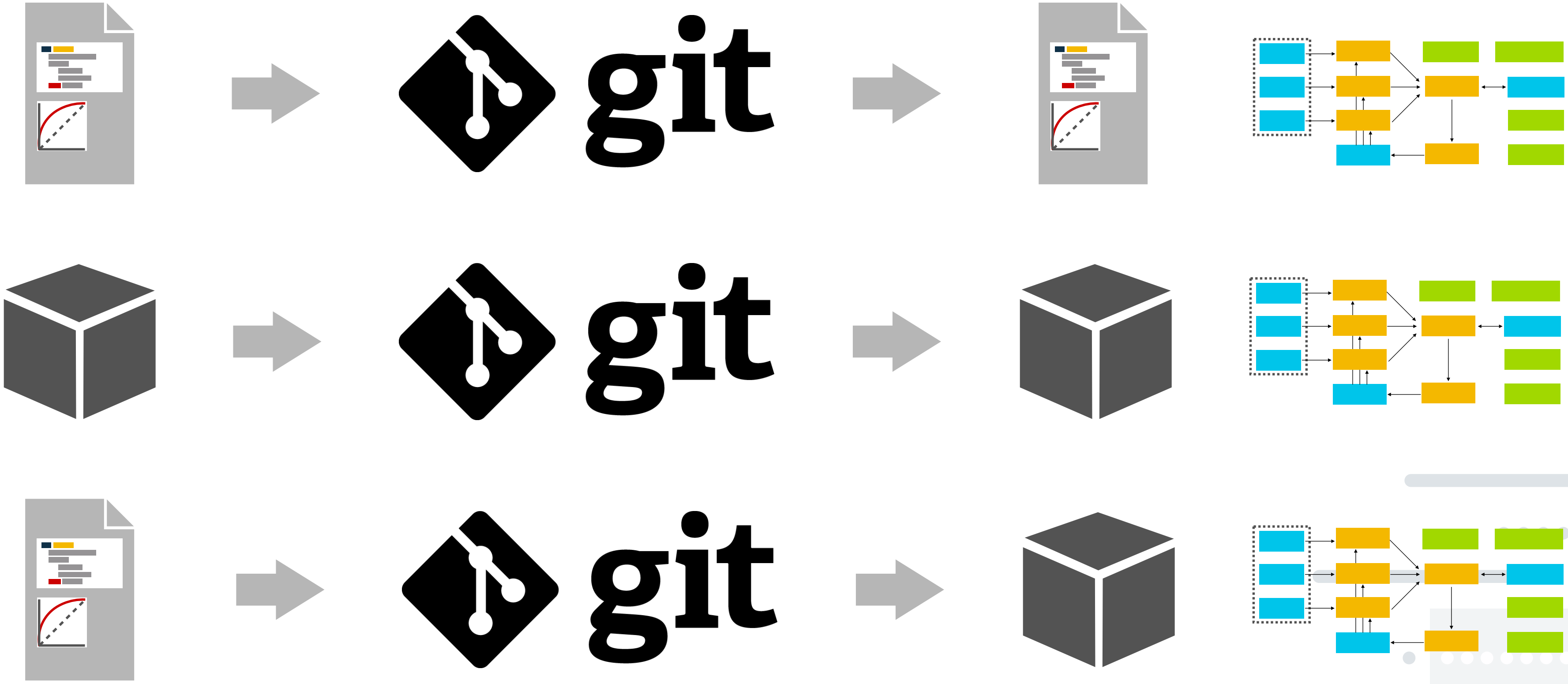
Integration and deployment



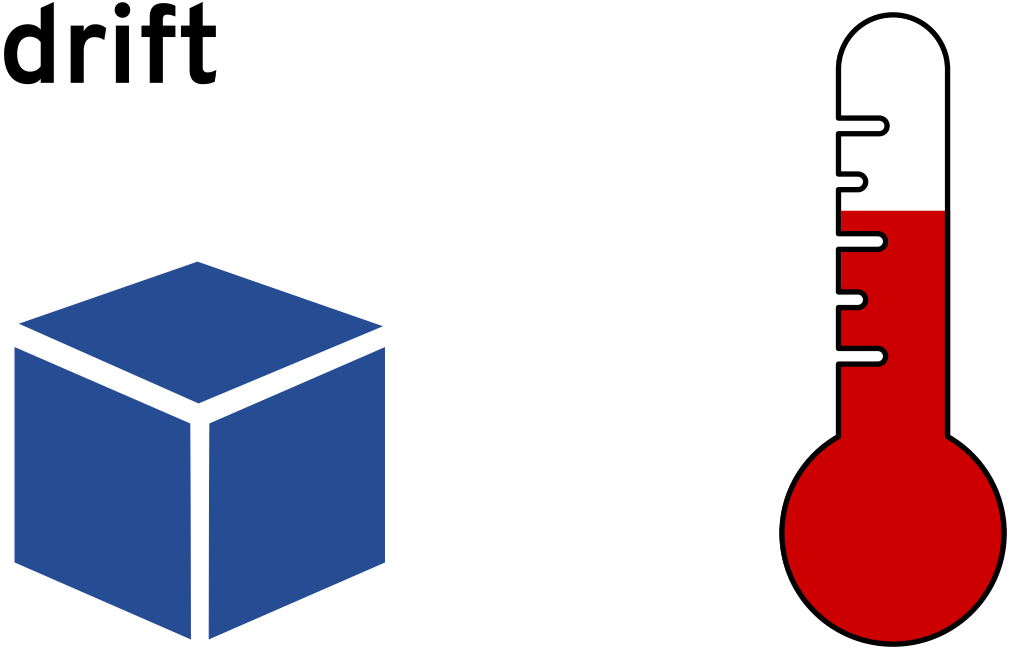
application code

configuration and installation recipes

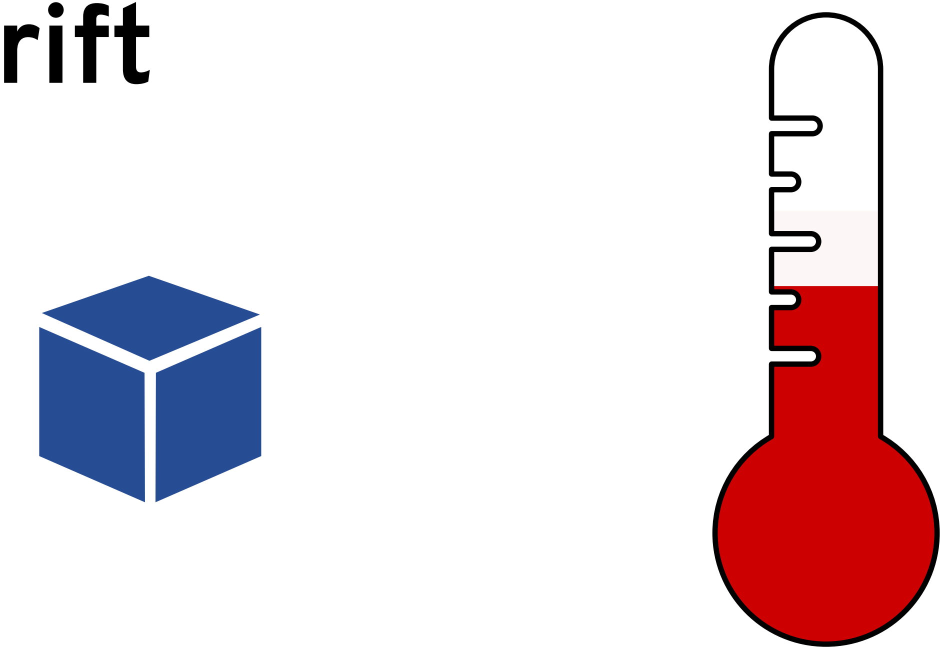
base image


git
git
git

Data drift



Data drift



Looking forward

Other frameworks

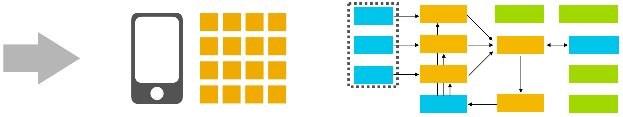
APACHE
Spark™

radanalytics.io

APACHE Spark™

jupyter

git



Open Data Hub

Kubeflow

jupyter

PyTorch

Other technologies

$$\mathbf{x} \cdot \mathbf{y} = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$$

Other technologies

$$\mathbf{x} \cdot \mathbf{y} = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$$

```
def dot(xs, ys):
  return sum([x * y for x, y in zip(xs, ys)])
```

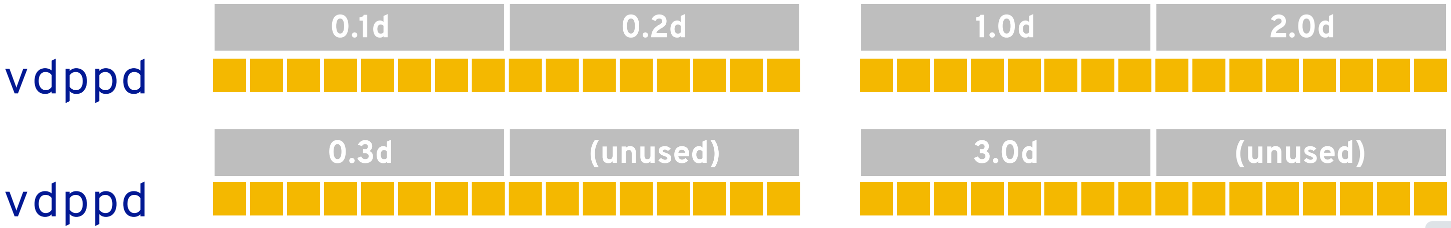
Other technologies

$$\mathbf{x} \cdot \mathbf{y} = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$$

```
def dot(xs, ys):
  return sum([x * y for x, y in zip(xs, ys)])

dot([0.1, 0.2, 0.3], [1.0, 2.0, 3.0])
```

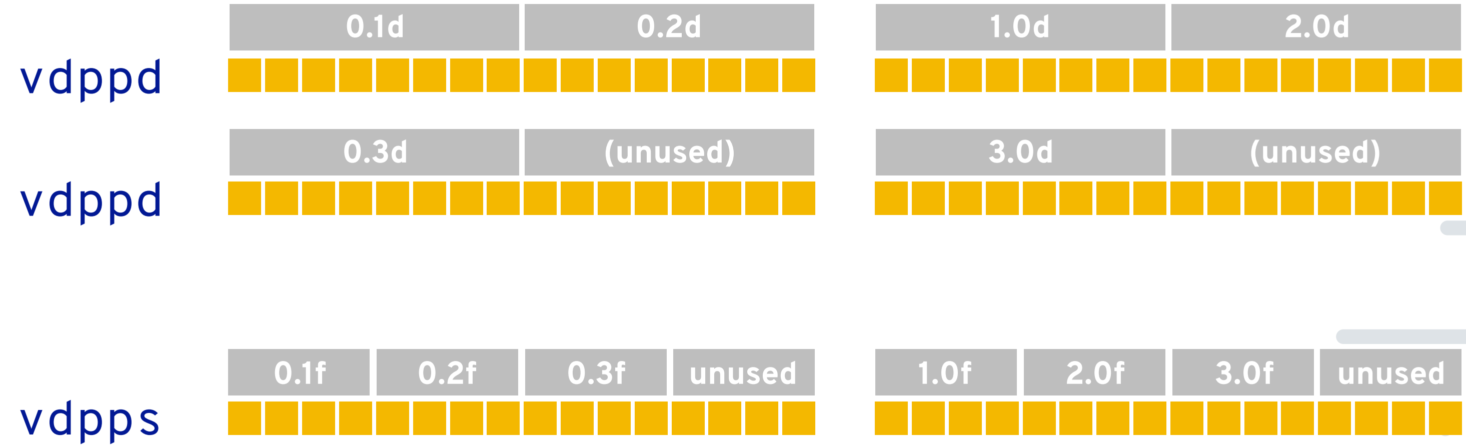
```
def dot(xs, ys):
  return sum([x * y for x, y in zip(xs, ys)])
```



vdppd 0.1d 0.2d 1.0d 2.0d

vdppd 0.3d (unused) 3.0d (unused)

```
def dot(xs, ys):
  return sum([x * y for x, y in zip(xs, ys)])
```

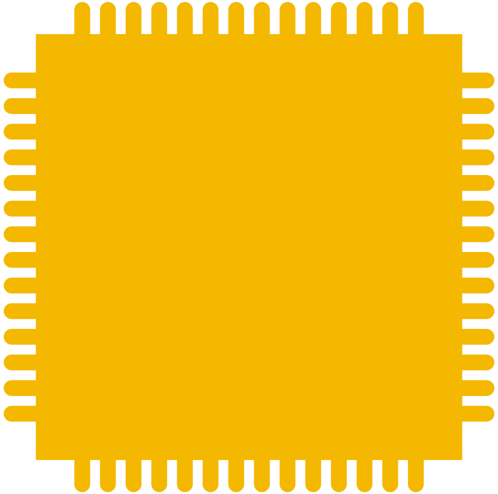


vdppd 0.1d 0.2d 1.0d 2.0d

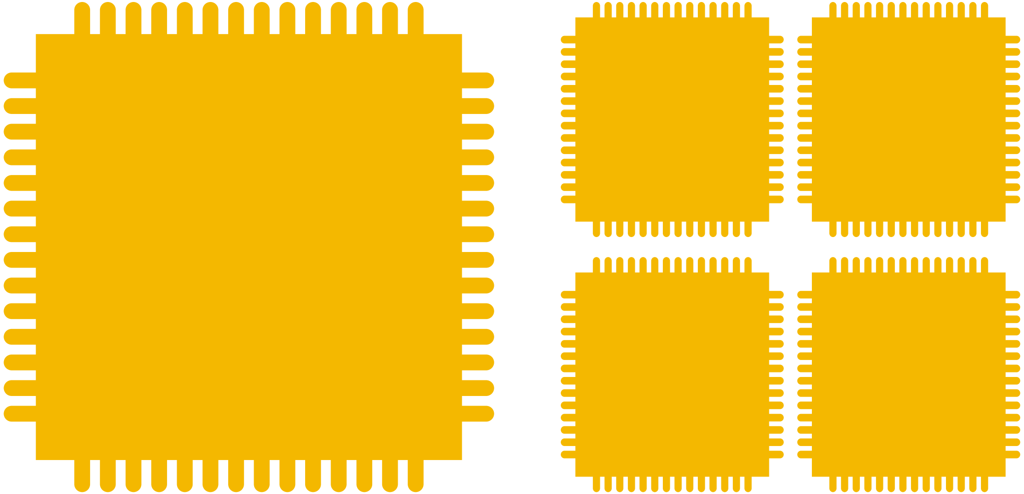
vdppd 0.3d (unused) 3.0d (unused)

vdpps 0.1f 0.2f 0.3f unused 1.0f 2.0f 3.0f unused

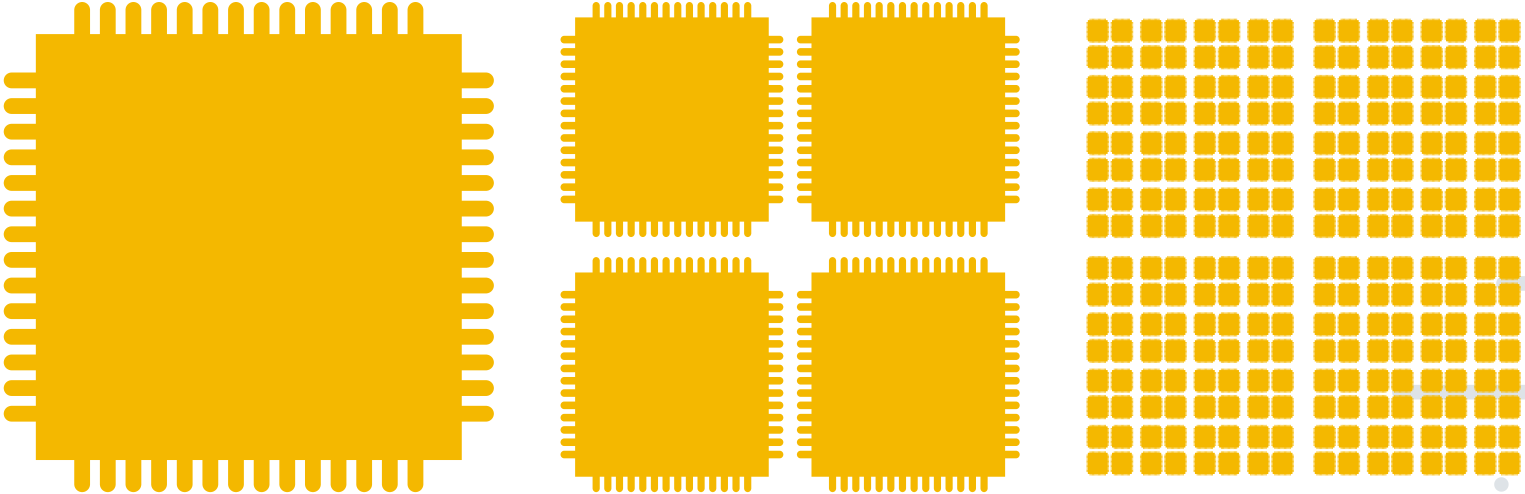
Other technologies



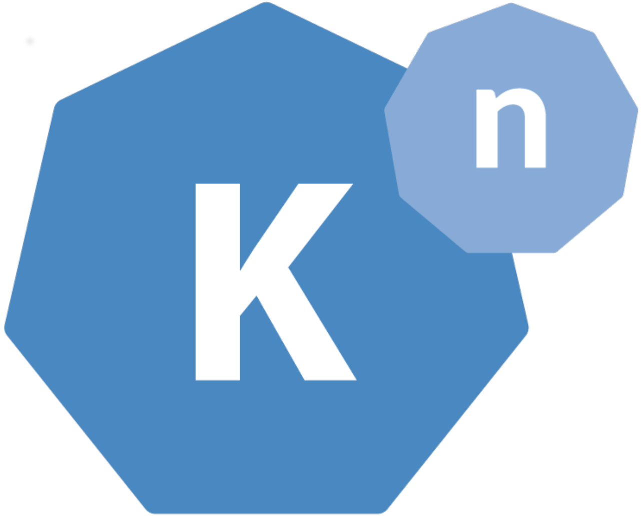
Other technologies



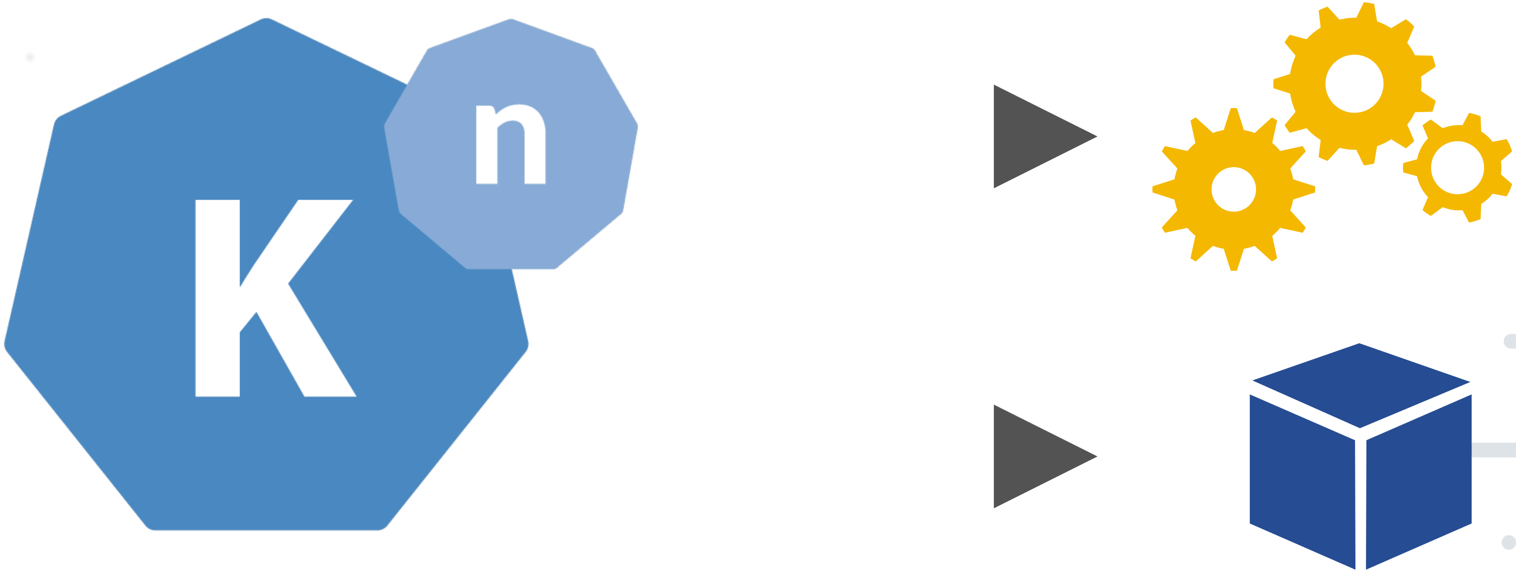
Other technologies



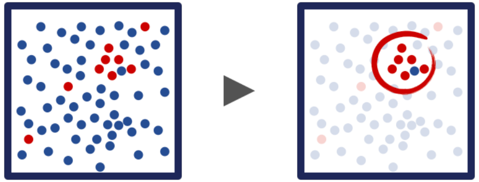
Other technologies

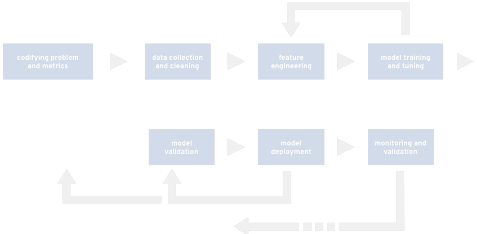


Other technologies



Conclusions





codifying problem and metrics

data collection and cleaning

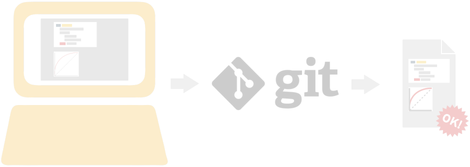
feature engineering

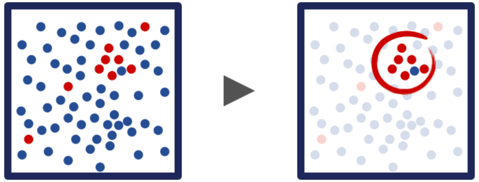
model training and tuning

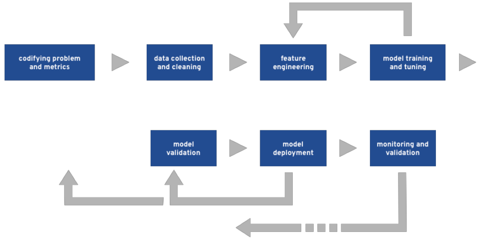
model validation

model deployment

monitoring and validation







codifying problem and metrics

data collection and cleaning

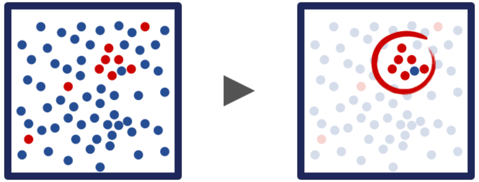
feature engineering

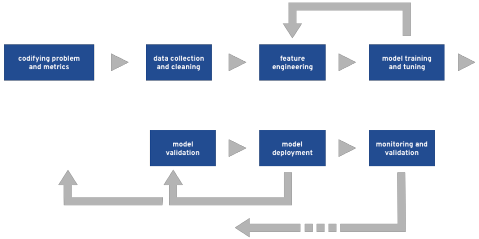
model training and tuning

model validation

model deployment

monitoring and validation





codifying problem and metrics

data collection and cleaning

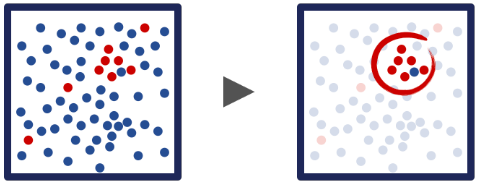
feature engineering

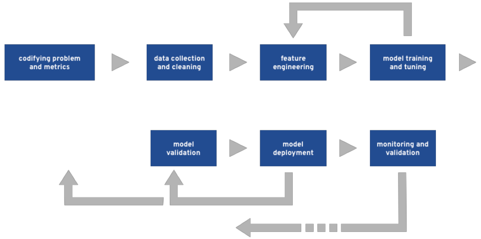
model training and tuning

model validation

model deployment

monitoring and validation





codifying problem and metrics

data collection and cleaning

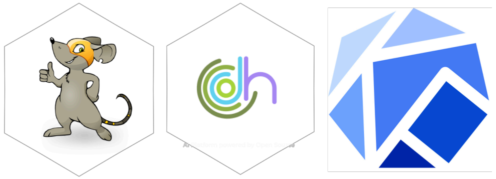
feature engineering

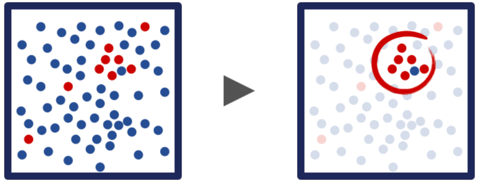
model training and tuning

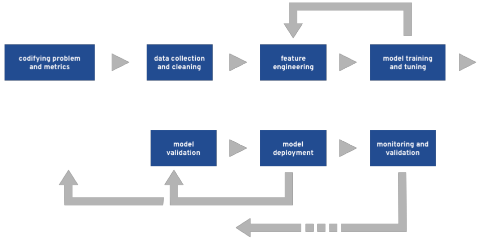
model validation

model deployment

monitoring and validation







codifying problem and metrics

data collection and cleaning

feature engineering

model training and tuning

model validation

model deployment

monitoring and validation

THANKS!

@sophwats • @willb • @FOSSJunkie