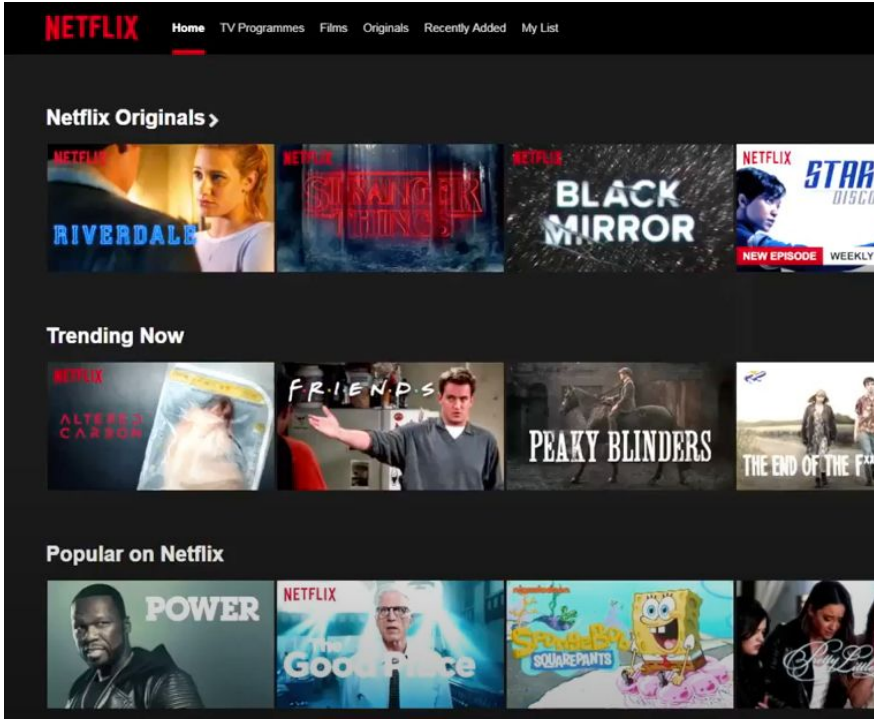
How Metaflow became a beloved Data Science framework at Netflix

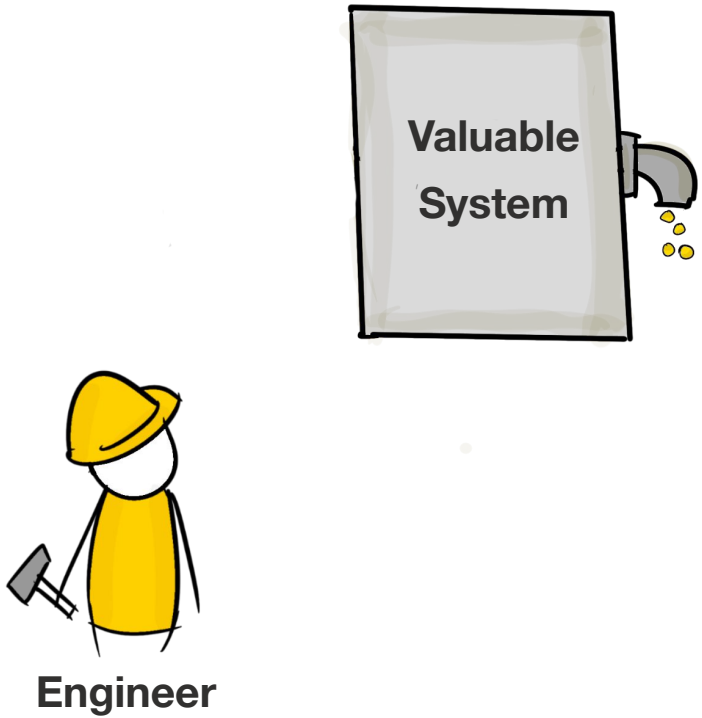
Julie Amundson
Machine Learning Infrastructure Leader

My first day at Netflix!



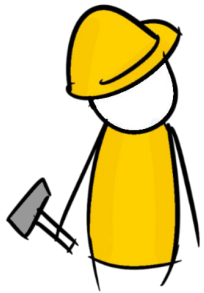
2008 vs. 2018

2008: Traditional software

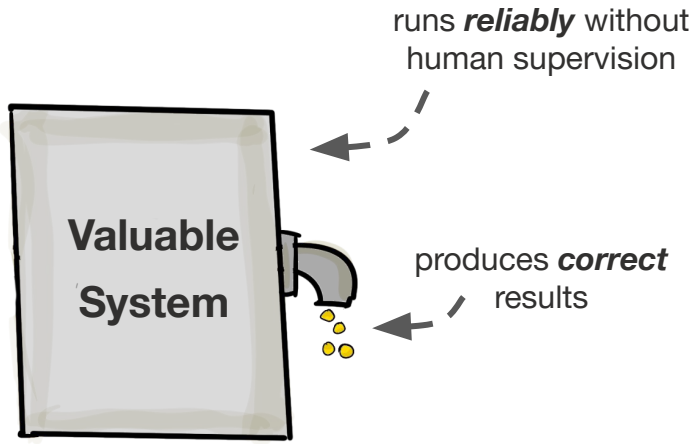


Slide courtesy of Ville Tuulos

2008: Traditional software



Engineer



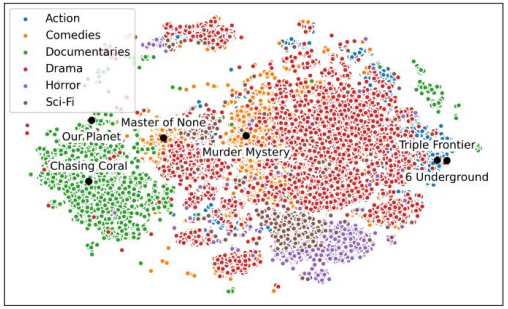
Slide courtesy of Ville Tuulos

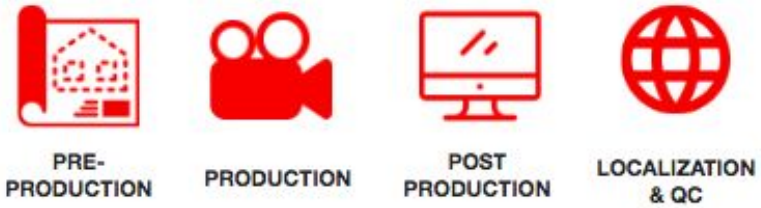
2008: I worked on the services behind the “play” button

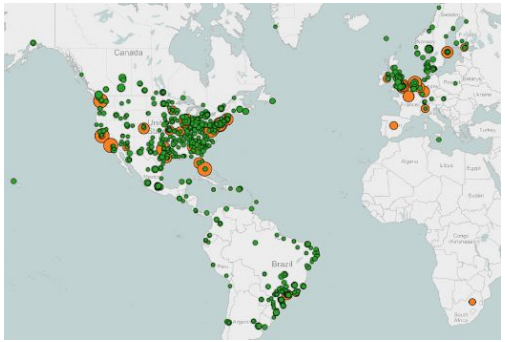
Me

Slide courtesy of Ville Tuulos

2018: Netflix rapidly expands ML investment







Content

$12B budget
& growing

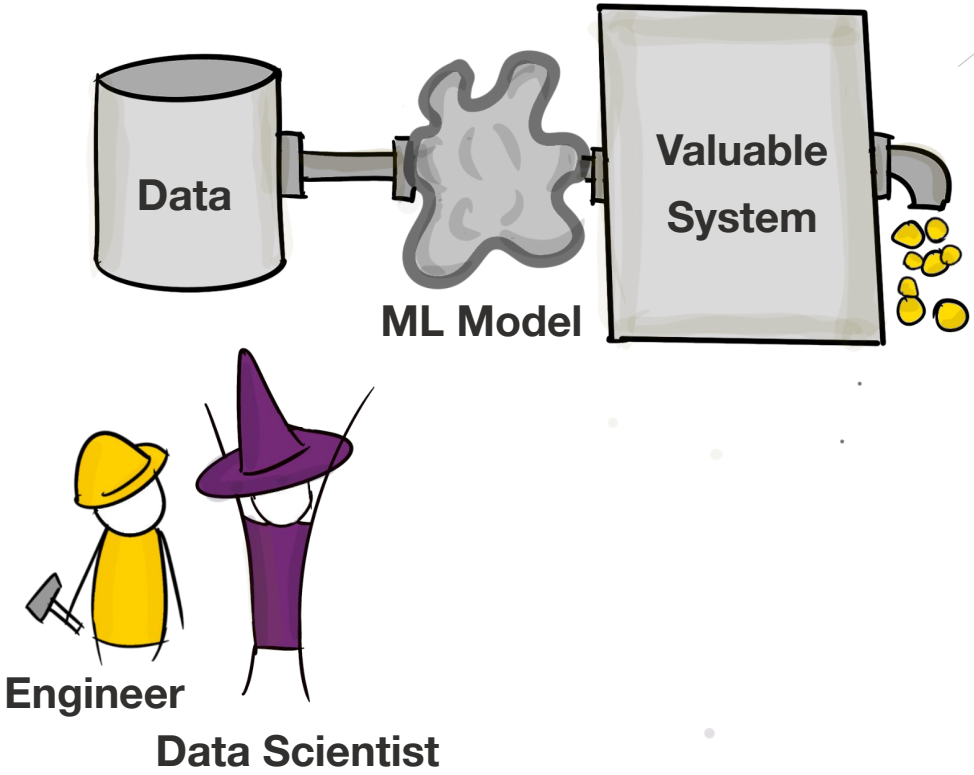
Studio

100s of concurrent productions

Product

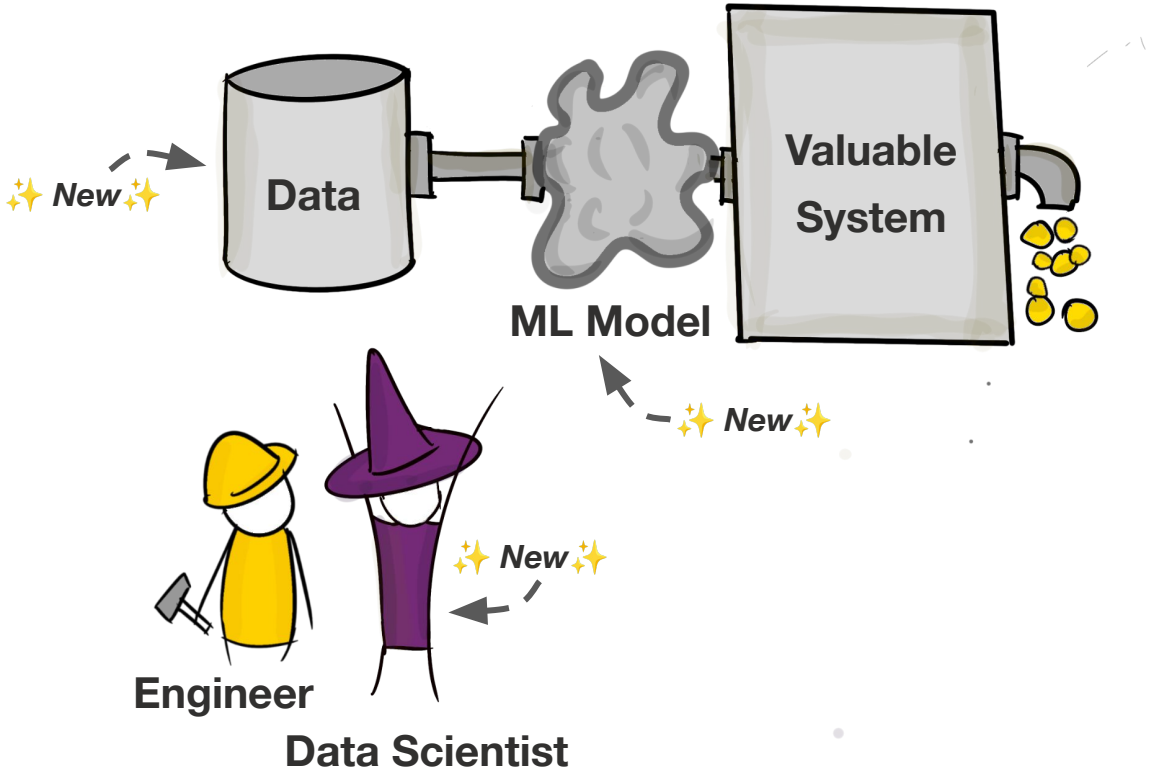
deliver the best experience for **220M+ members**

2018: ML-powered software



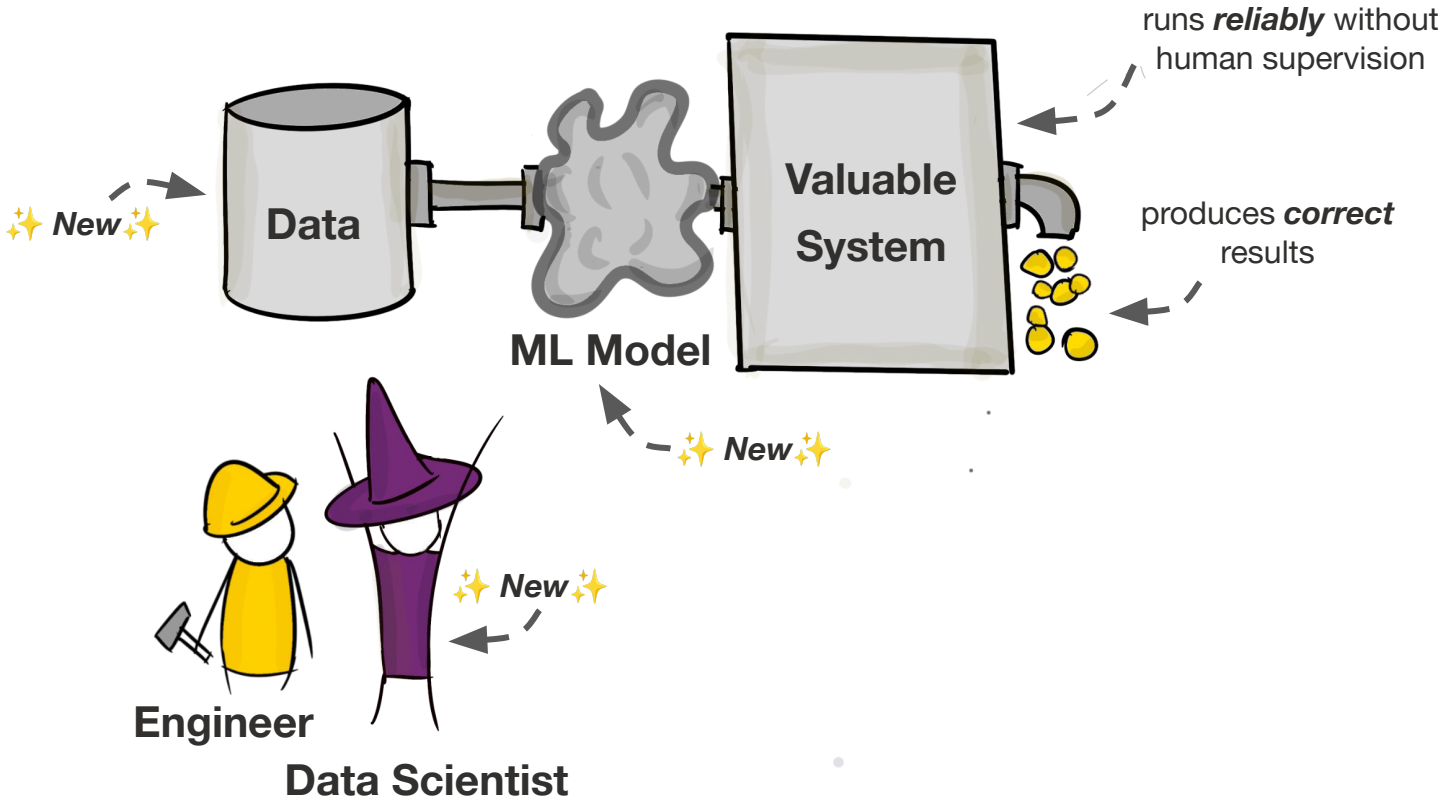
Slide courtesy of Ville Tuulos

2018: ML-powered software



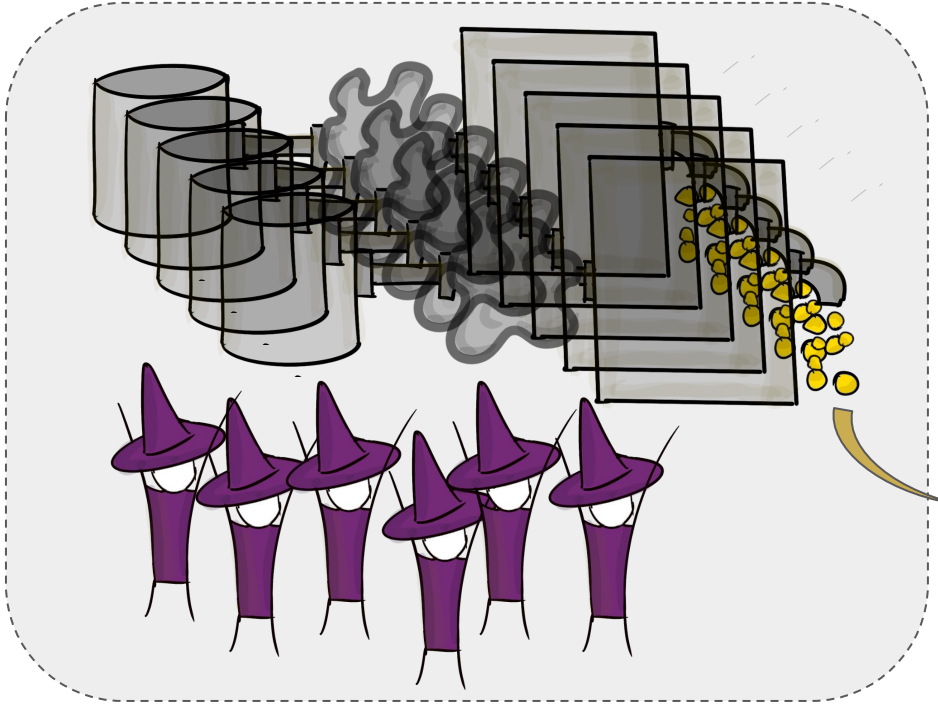
Slide courtesy of Ville Tuulos

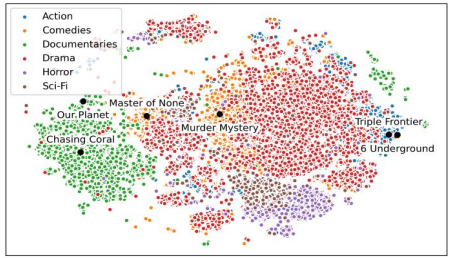
2018: ML-powered software

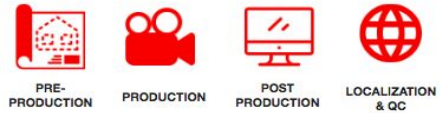


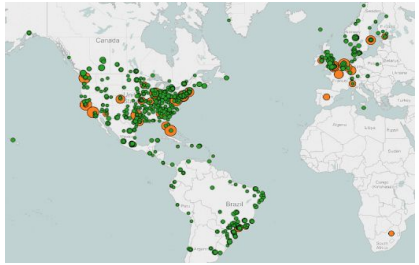
Slide courtesy of Ville Tuulos

Netflix needed to produce many impactful ML-powered applications





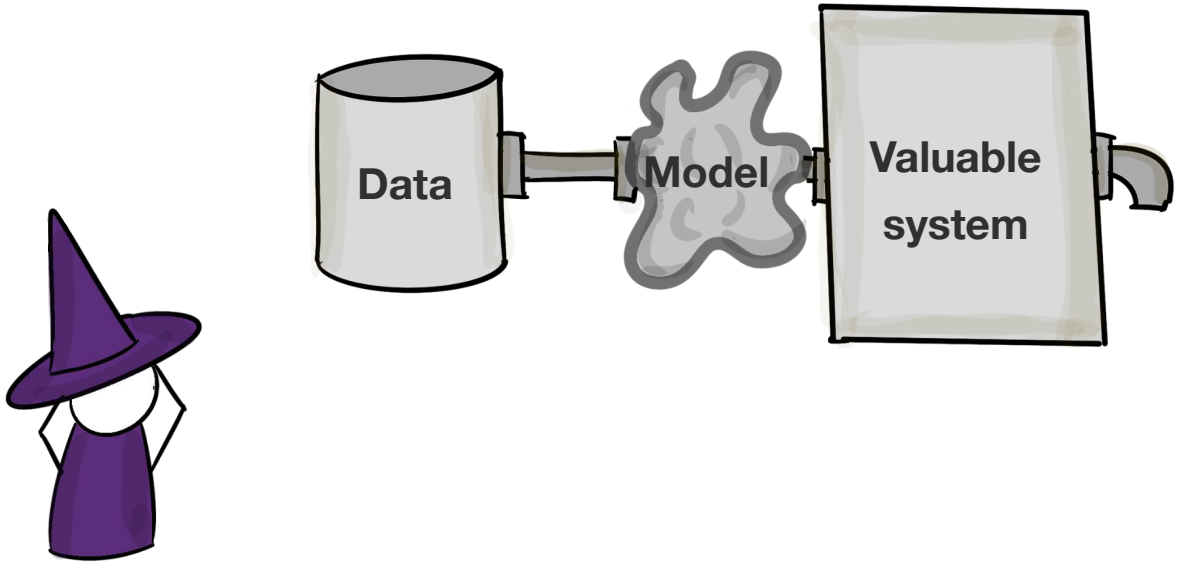




?

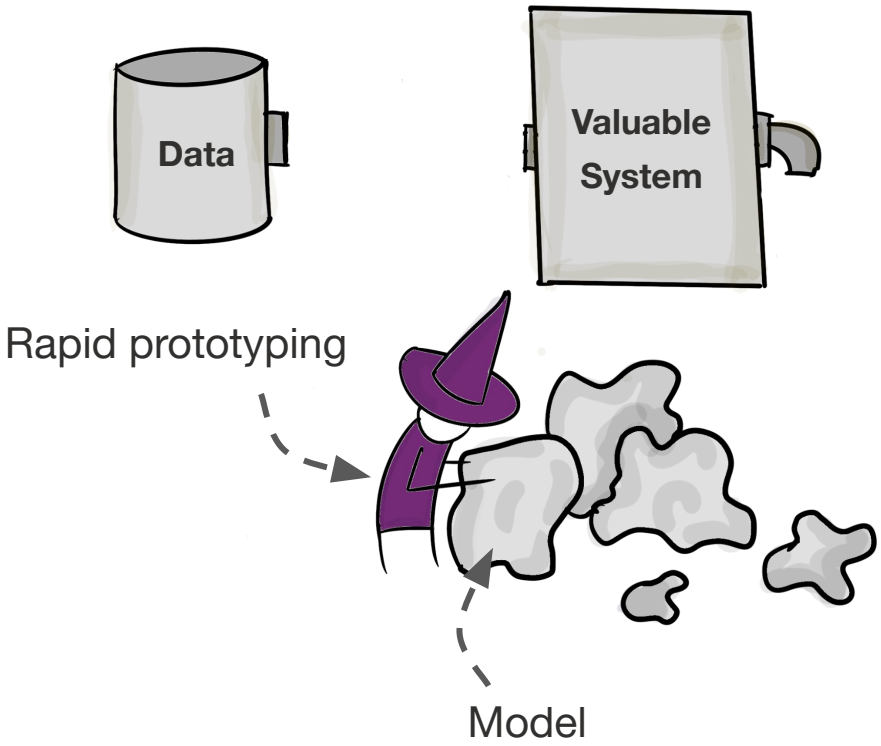
Illustrations courtesy of Ville Tuulos

Sonia is building sentiment analysis models for Marketing



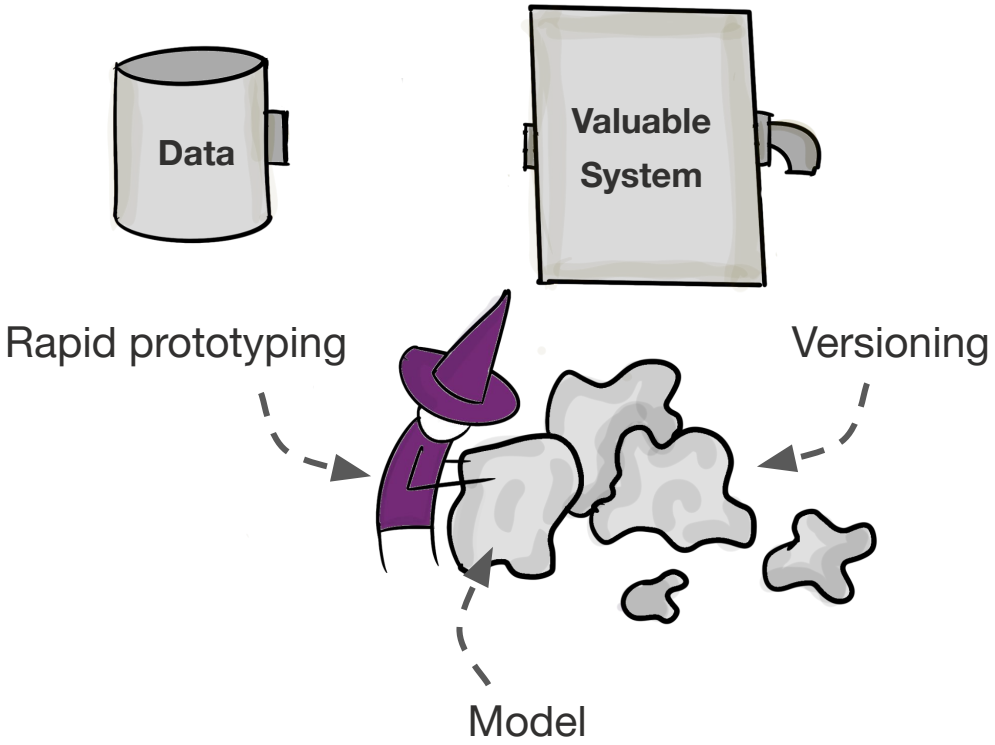
Illustrations courtesy of Ville Tuulos

First, she needs to build a prototype



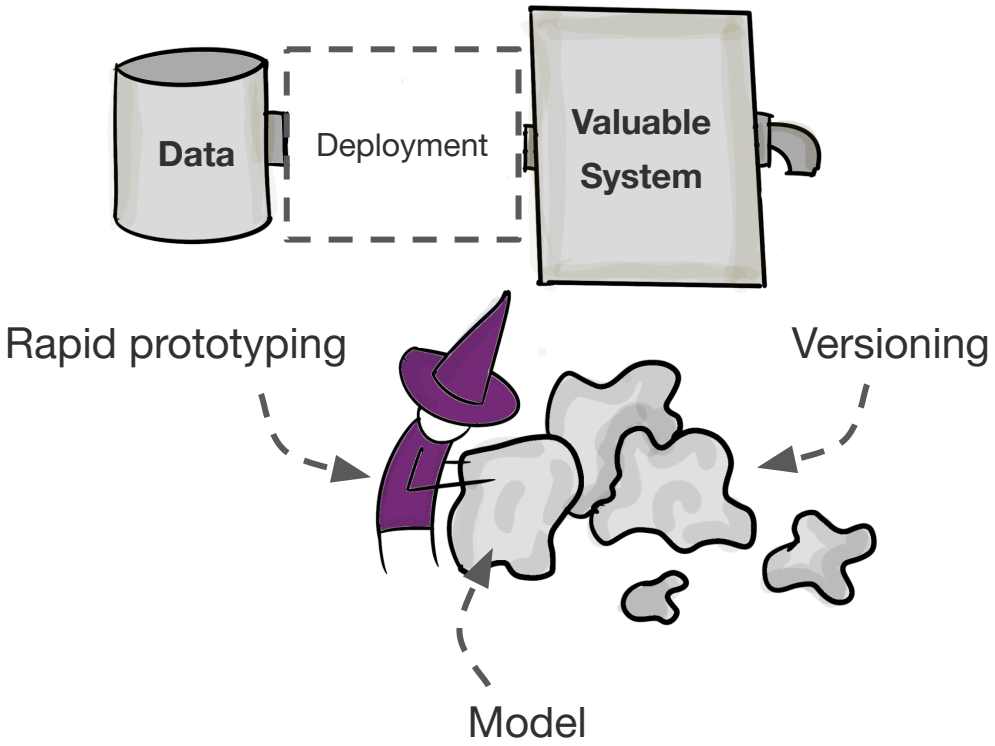
Slide courtesy of Ville Tuulos

Sonia needs to iterate on many ideas to find a prototype that works



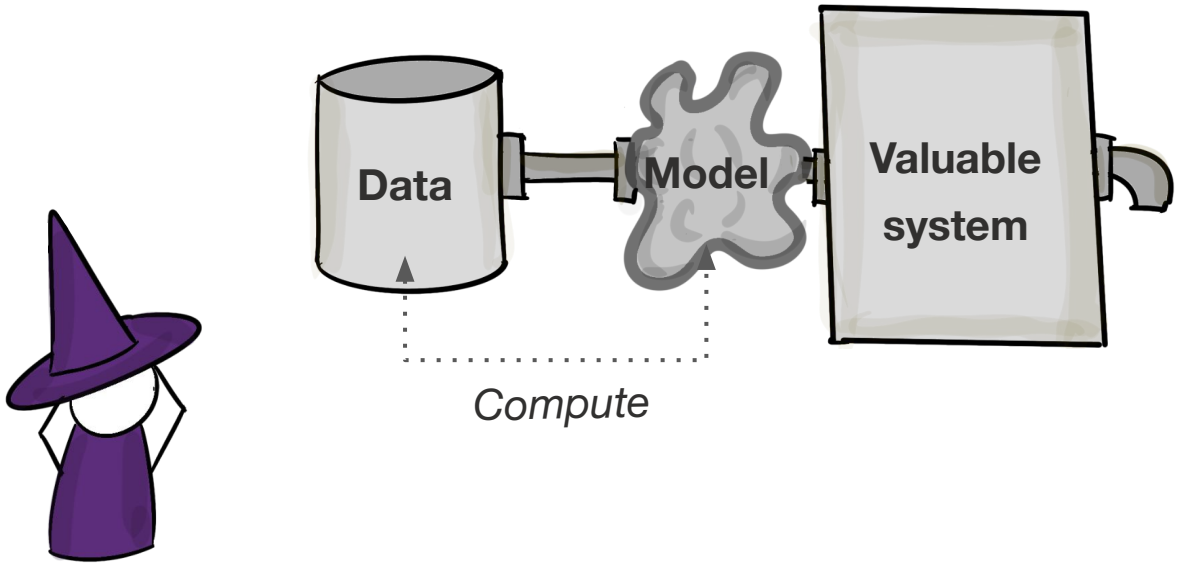
Slide courtesy of Ville Tuulos

She needs a way to reliably deploy to production



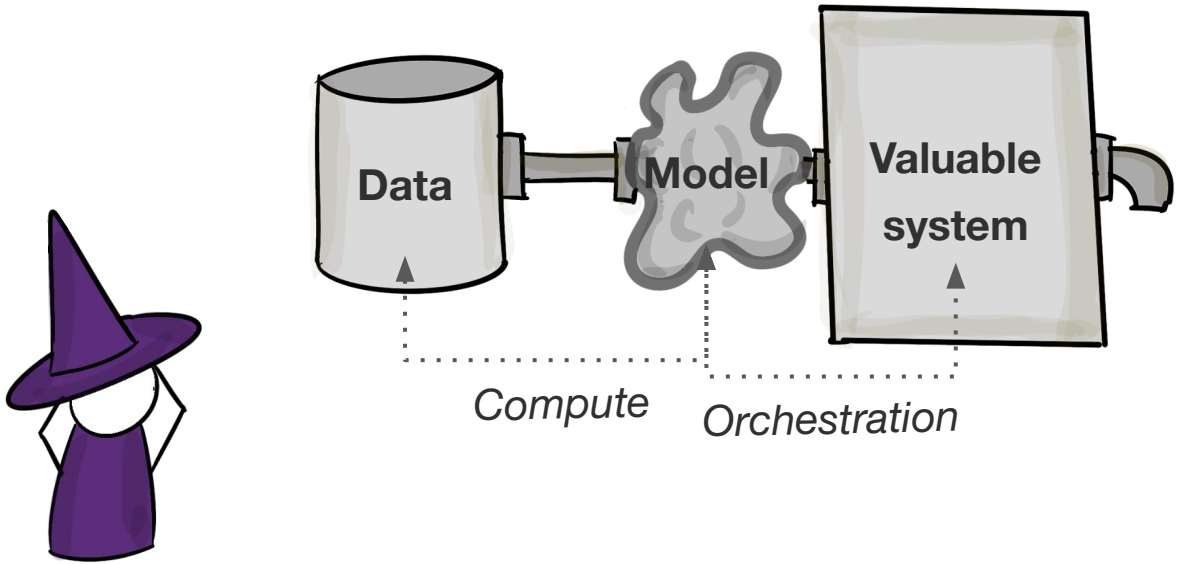
Slide courtesy of Ville Tuulos

Model training and scoring require compute



Illustrations courtesy of Ville Tuulos

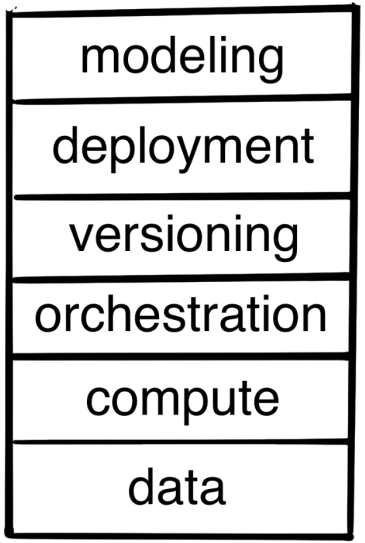
Sonia's modeling pipeline needs orchestration!



Illustrations courtesy of Ville Tuulos

Putting the pieces together

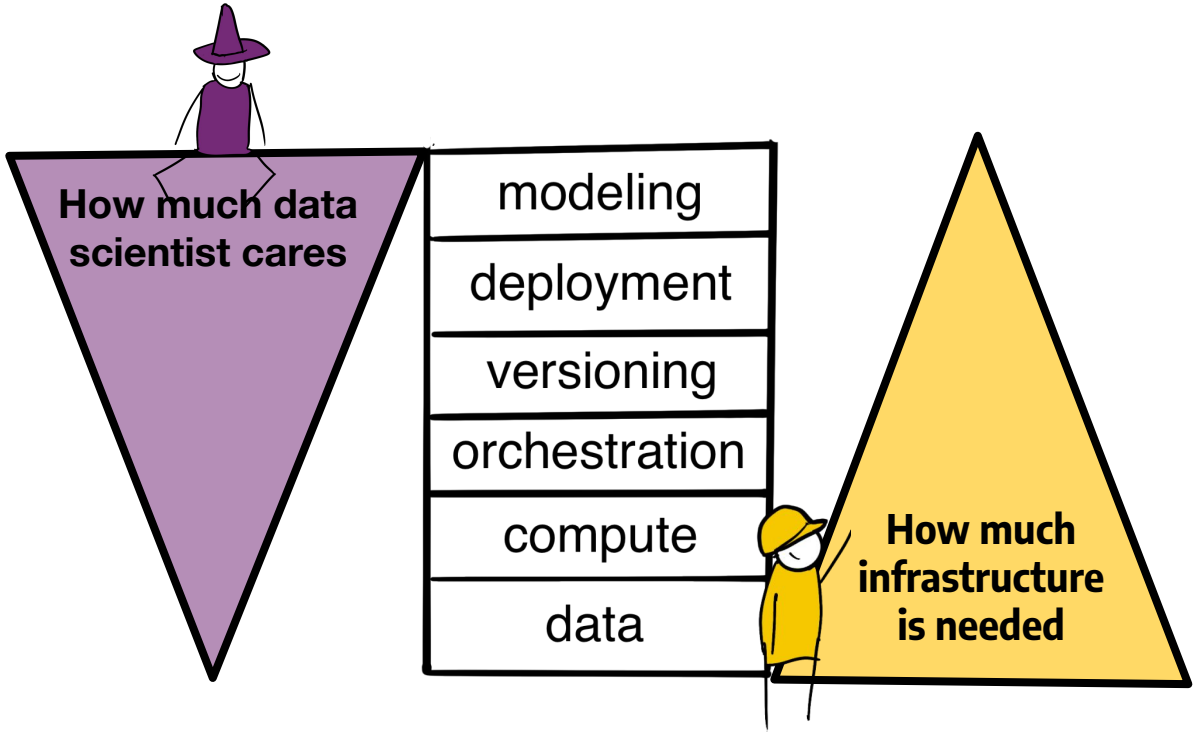
The ML Infrastructure Stack



Slide courtesy of Ville Tuulos

Putting the pieces together

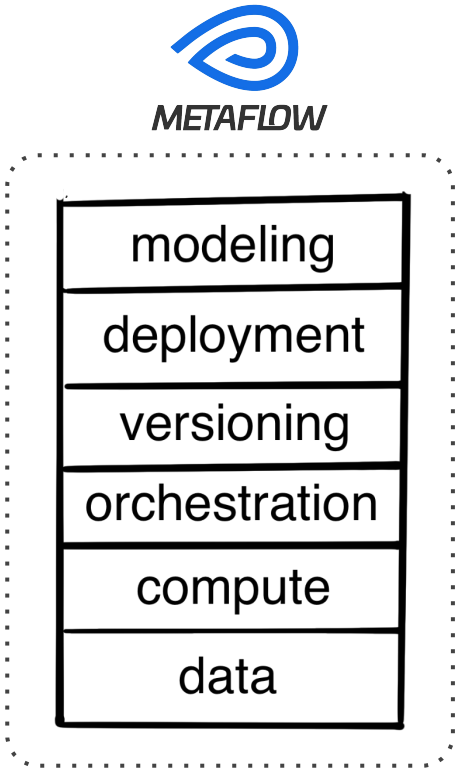
The ML Infrastructure Stack



Slide courtesy of Ville Tuulos

Metaflow

A full-stack, human-friendly framework for data science



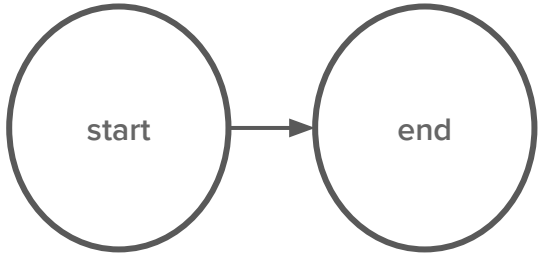
Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



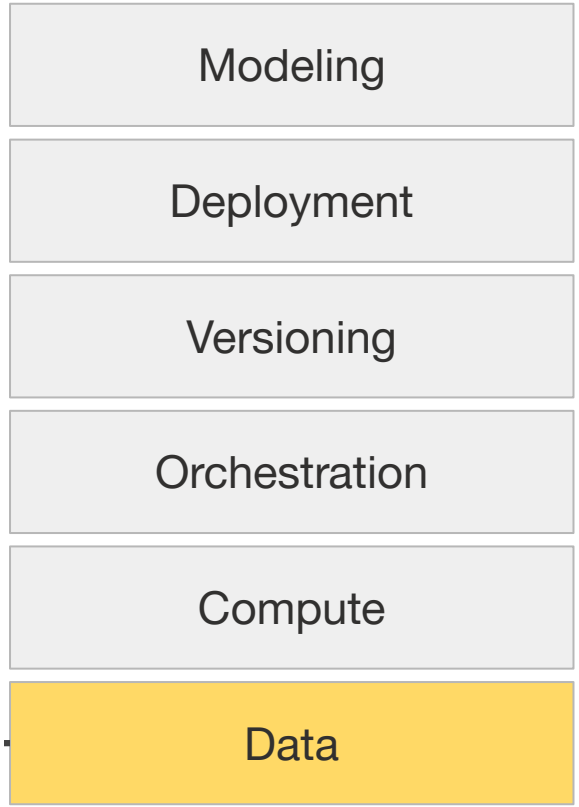
Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



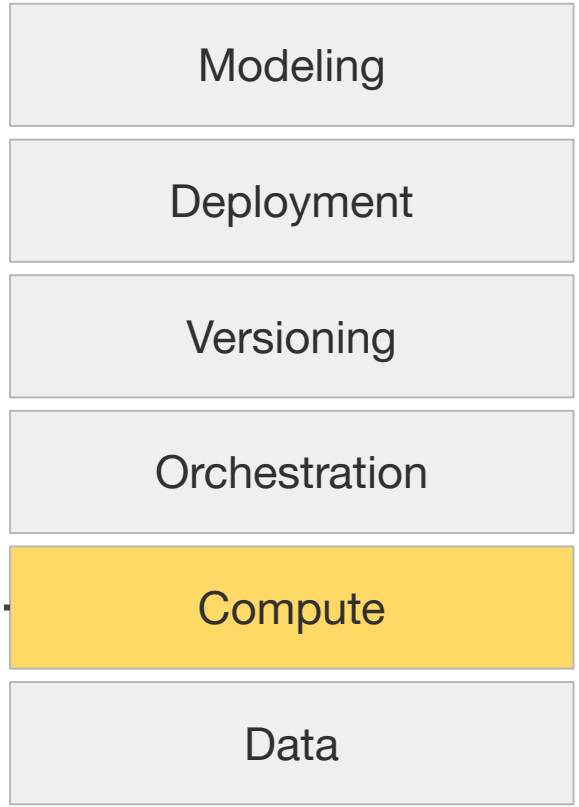
Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



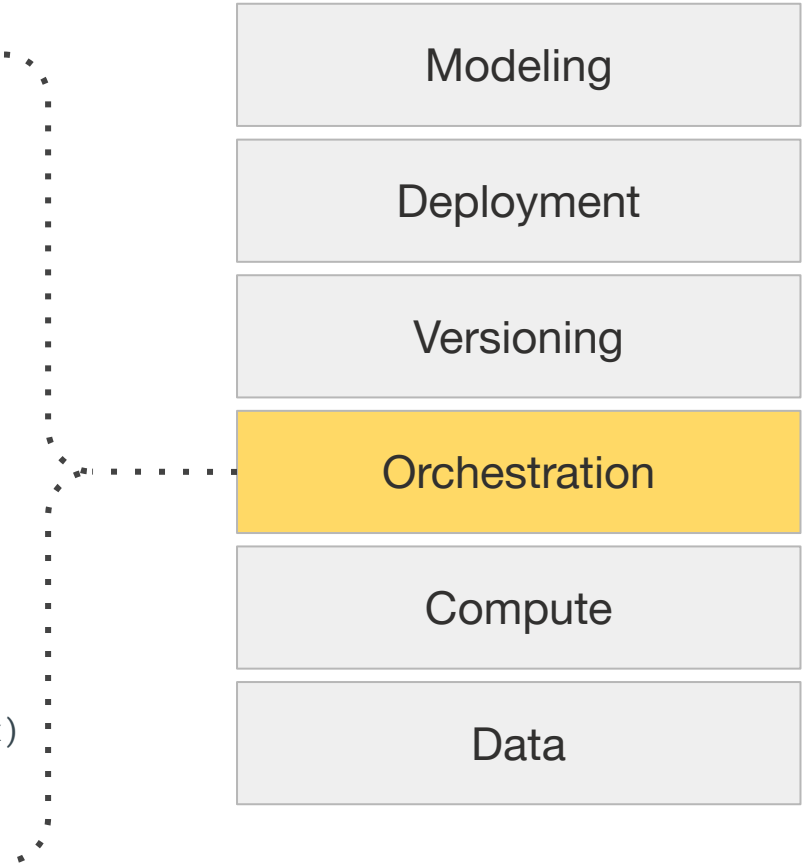
Slide courtesy of Ville Tuulos

```
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



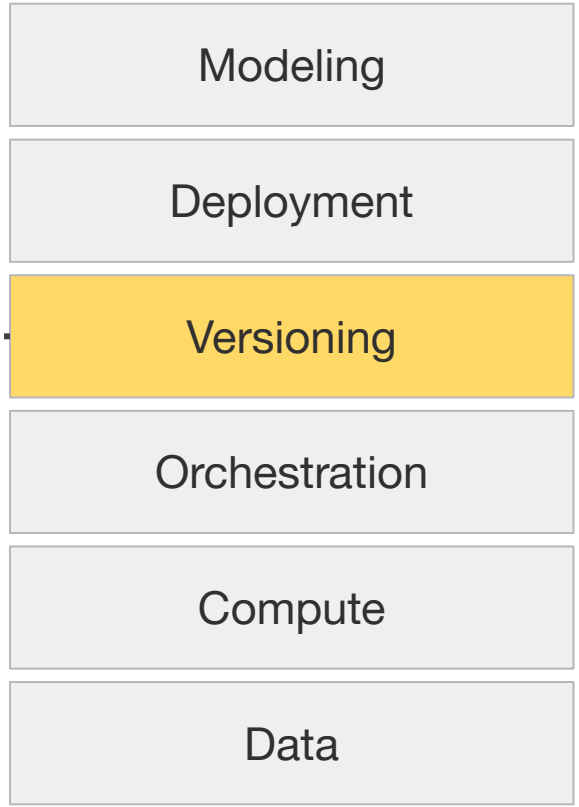
Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```

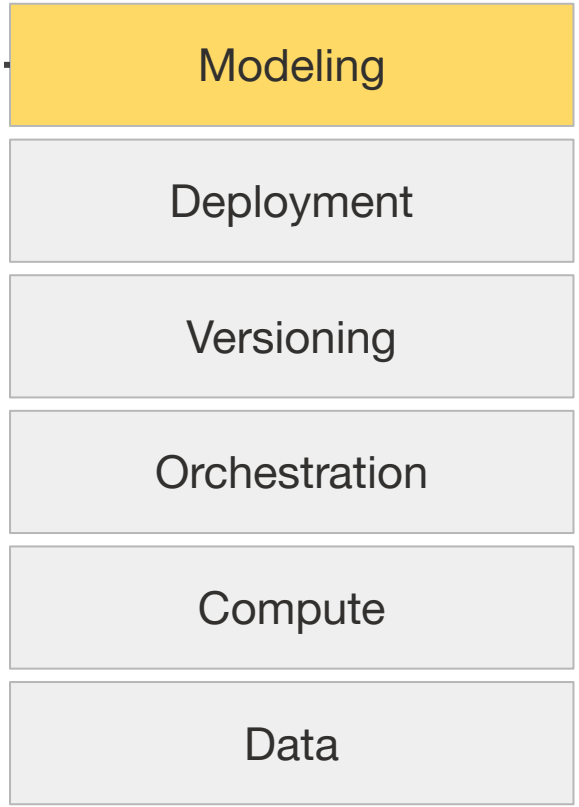
Slide courtesy of Ville Tuulos

```python
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```



Slide courtesy of Ville Tuulos

```
from metaflow import FlowSpec, step, conda_base,\
                     kubernetes, schedule

@conda_base(libraries={'scikit-learn': '1.1.2'})
@schedule(daily=True)
class HelloFlow(FlowSpec):
    @step
    def start(self):
        self.x = 1
        self.next(self.end)

    @kubernetes(memory=64000)
    @step
    def end(self):
        self.x += 1
        print("Hello world! The value of x is", self.x)

if __name__ == '__main__':
    HelloFlow()
```

The code may look nice but it doesn’t produce value by itself

Slide courtesy of Ville Tuulos

To produce real value, the code needs to integrate seamlessly *with the surrounding infrastructure*

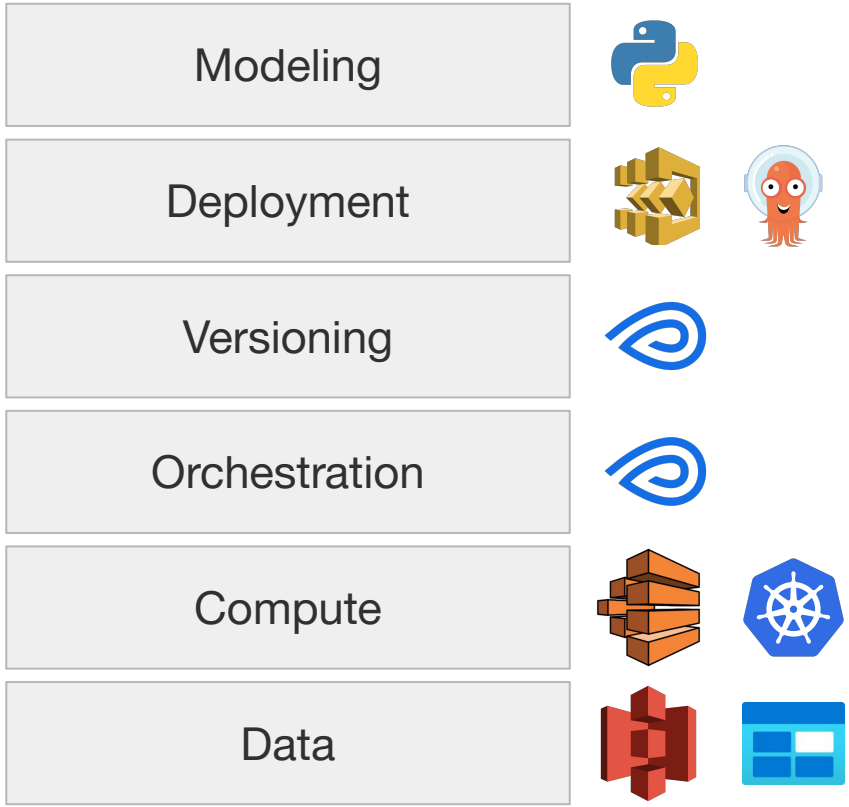


Diagram courtesy of Ville Tuulos

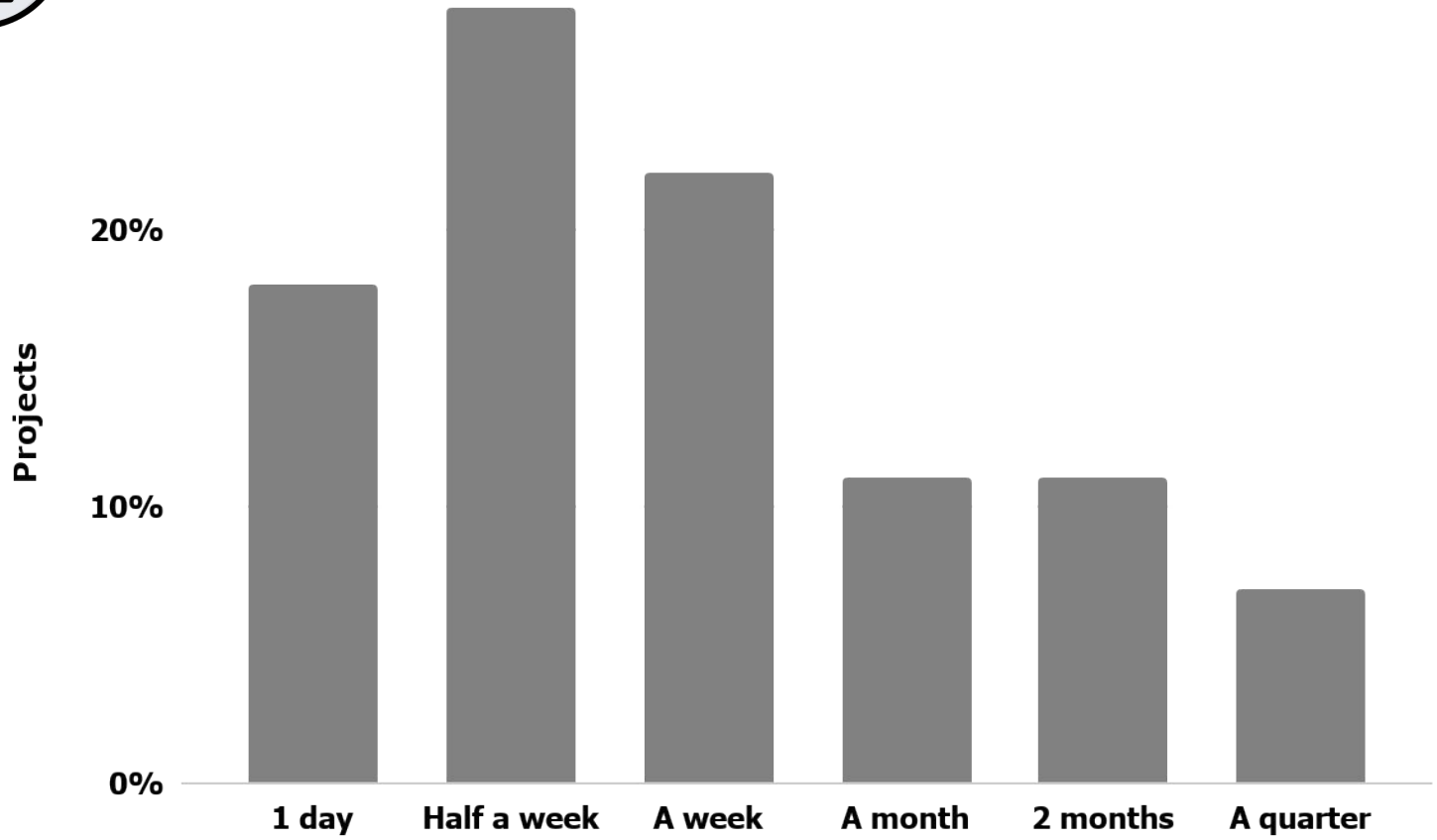
Metaflow impact at Netflix

Velocity

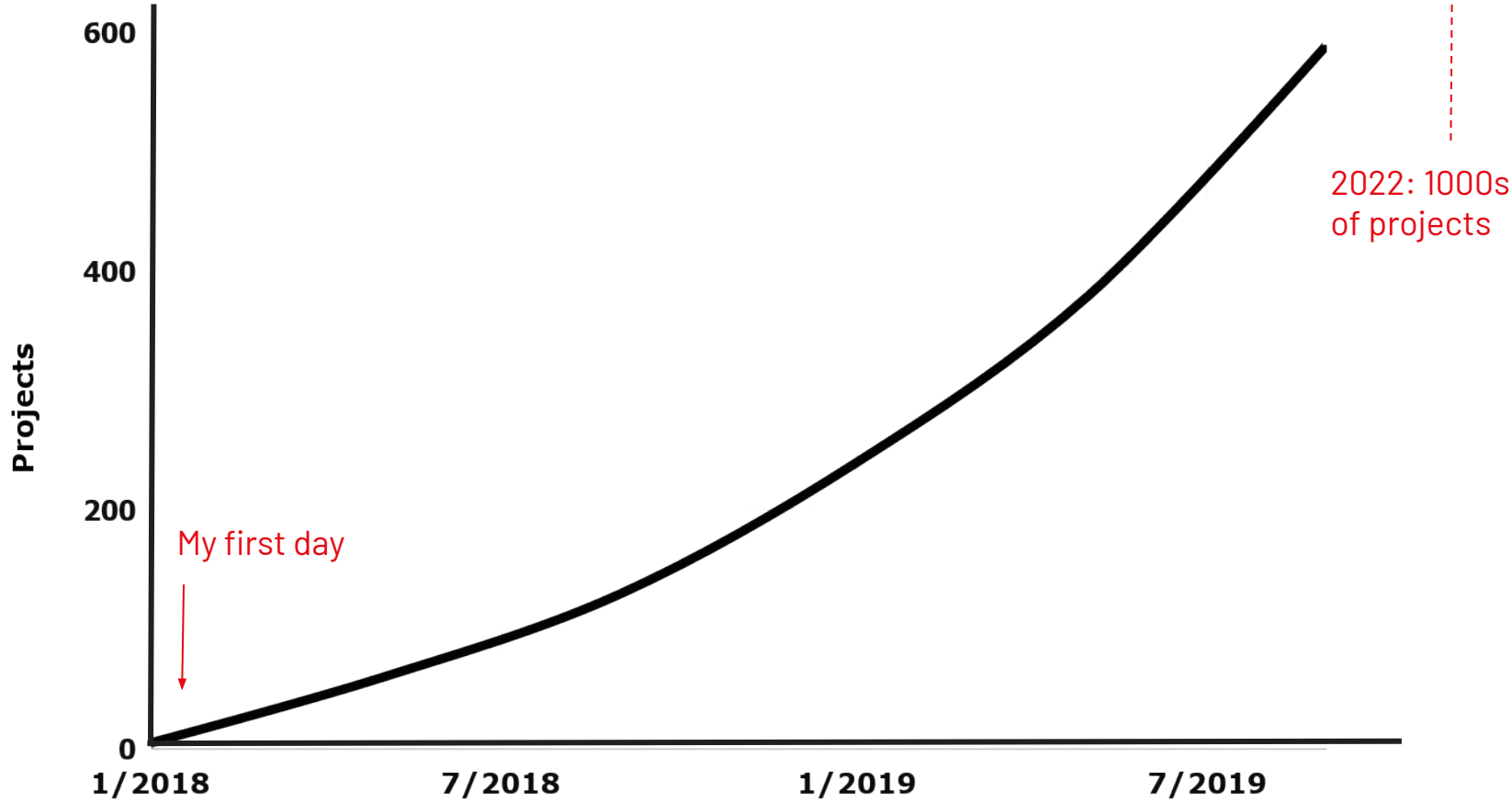
Volume

Variety

Velocity: Time to production



Volume: Metaflow Adoption



Beyond Variety: Netflix Open Source

1.8k+

community members

NETFLIX OSS **7th**

most popular project

aws

multiple cloud integrations

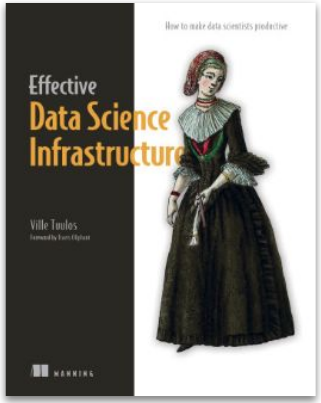
23andMe CNN realtor.com®

adopted by hundreds of companies

Want to learn more about Metaflow?

docs.metaflow.org

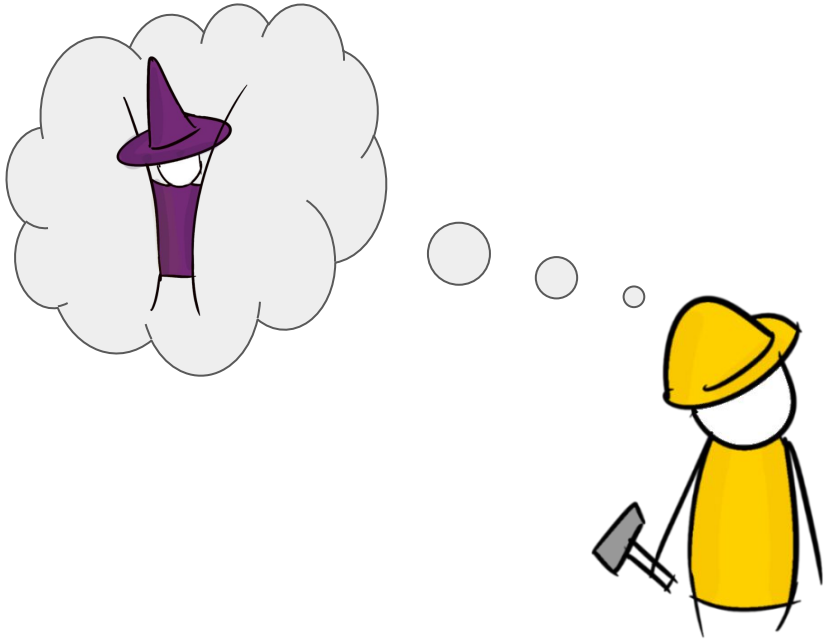
outerbounds.com



Effective Data Science Infrastructure

New book, by Ville Tuulos

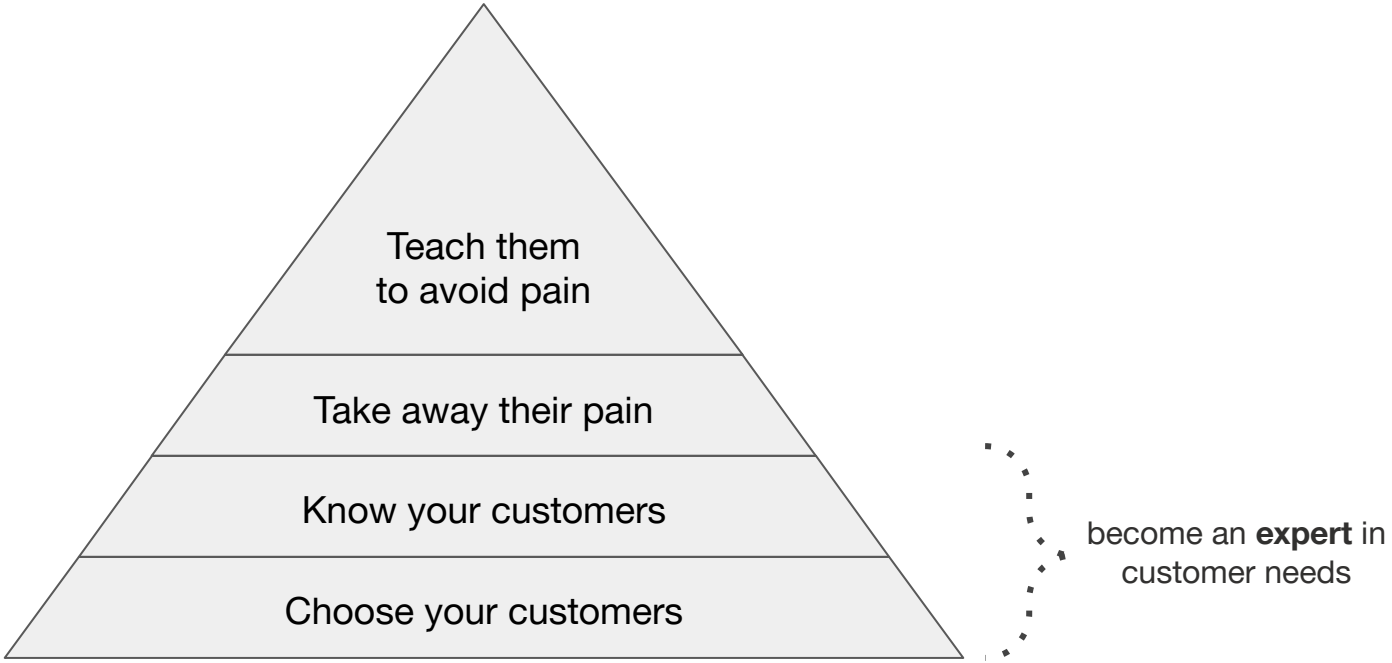
Metaflow's success began with customer obsession



Illustrations courtesy of Ville Tuulos

Customer obsession in action

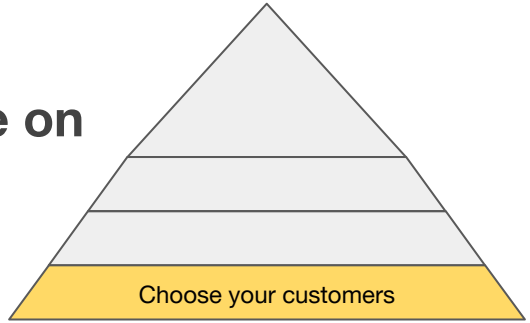
Customer obsession in action



Customer obsession in action

equip your customers
with **superpowers**

Choosing Data Scientists meant we could concentrate on their needs



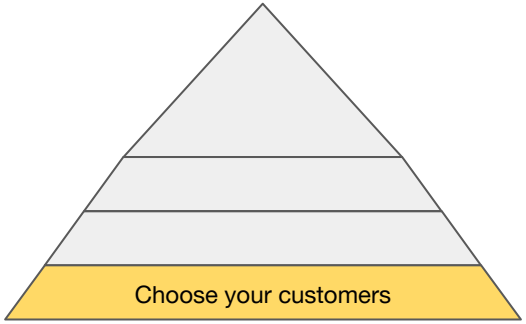
Customers

- Data Scientists
- ML Engineers

Non-Customers

- Algorithm Engineers
- Data Engineers
- Analytics Engineers
- Software Engineers

Choosing Data Scientists helped to clarify our product strategy

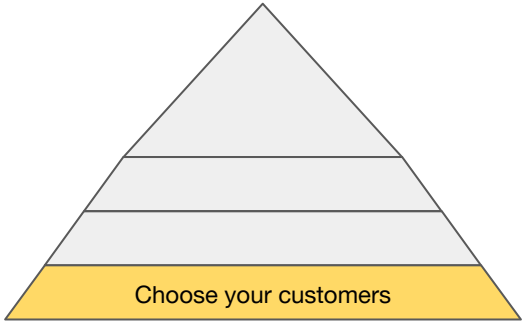


conventional wisdom

modeling
deployment
versioning
orchestration
compute
data

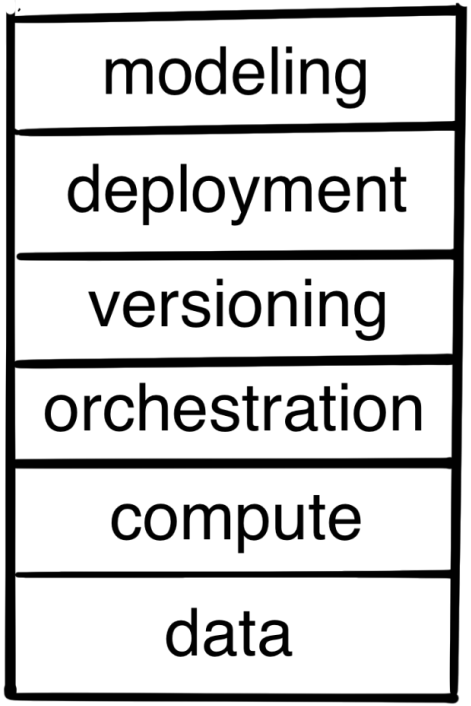
Metaflow

Choosing Data Scientists enabled us to focus



Non-features

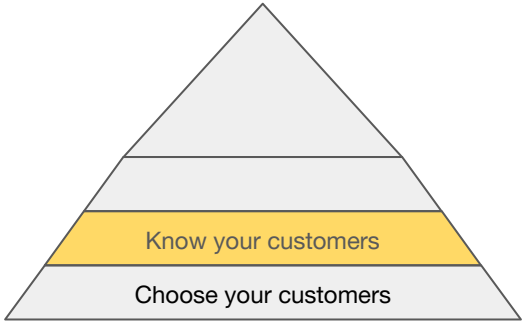
- Generic feature store
- Training framework
- Generic Model registry
- …etc…

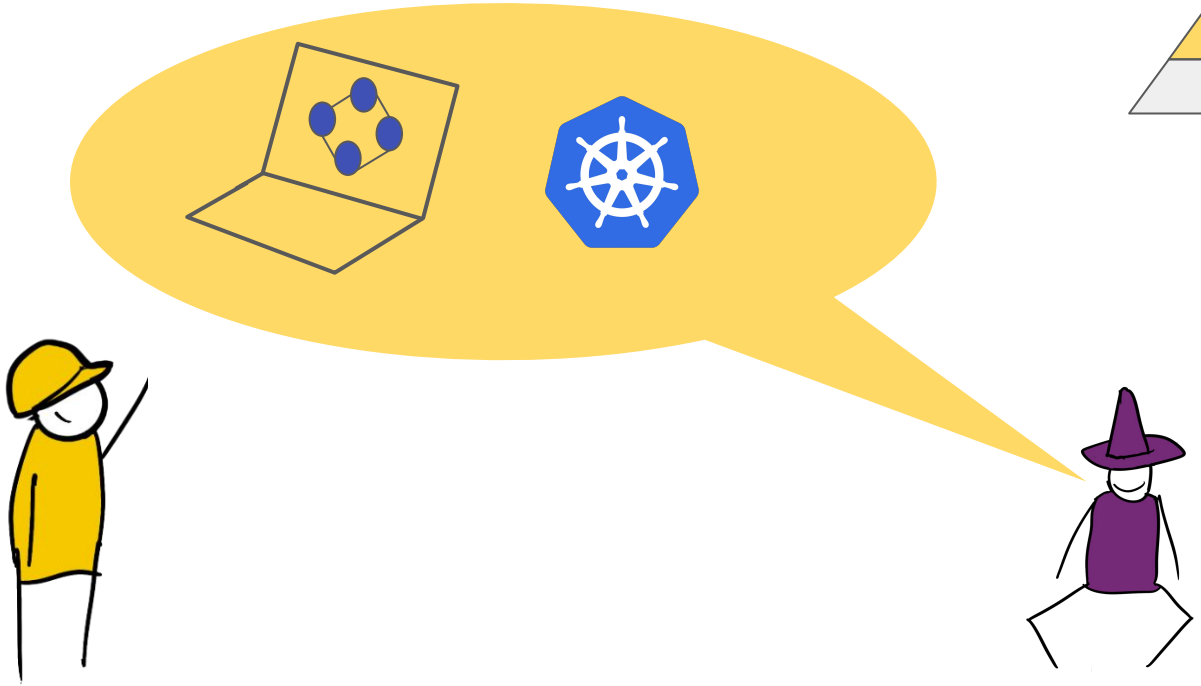


Features

- Data-parallel training
- Job scheduling
- Reproducibility
- …etc…

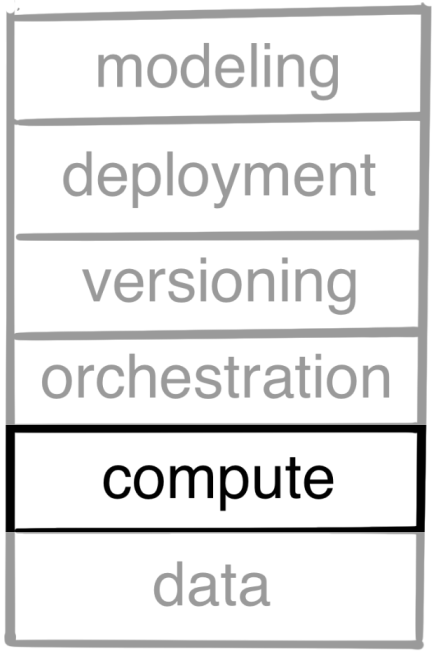
We spoke to many Data Scientists to learn about their daily lives, *in detail*

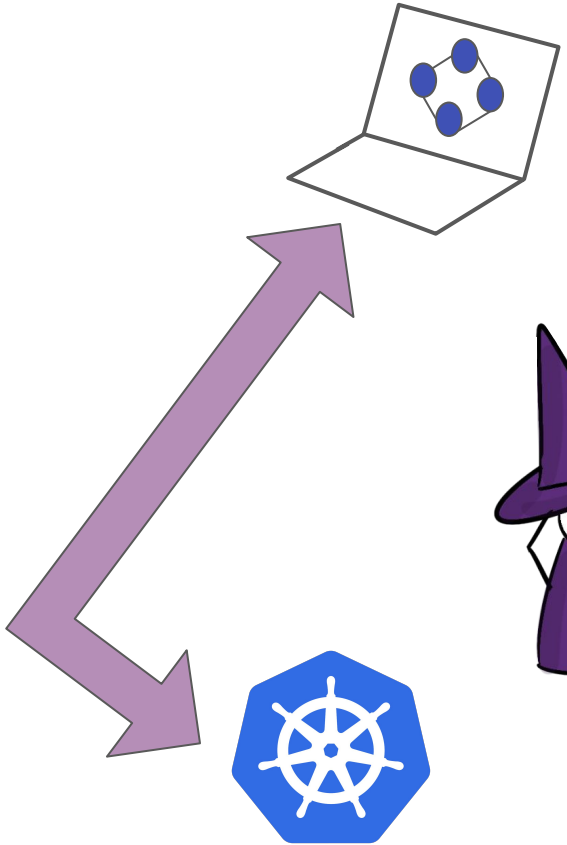


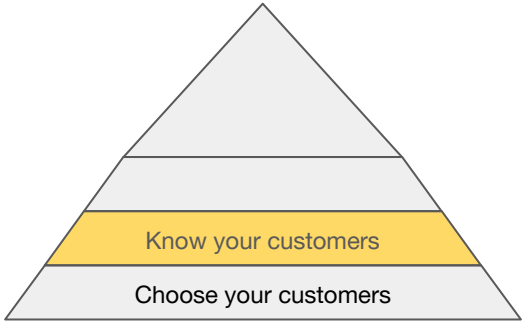


Further reading: https://manuelohan.com/wp-content/uploads/2017/05/The-Mom-Test-en.pdf

After many conversations, patterns emerged

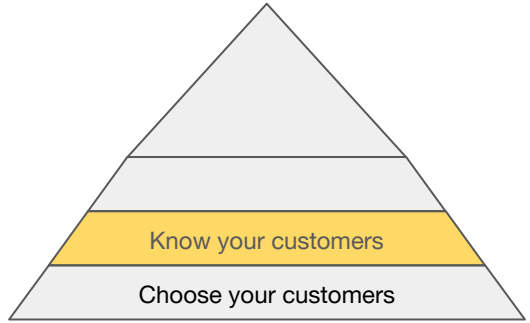


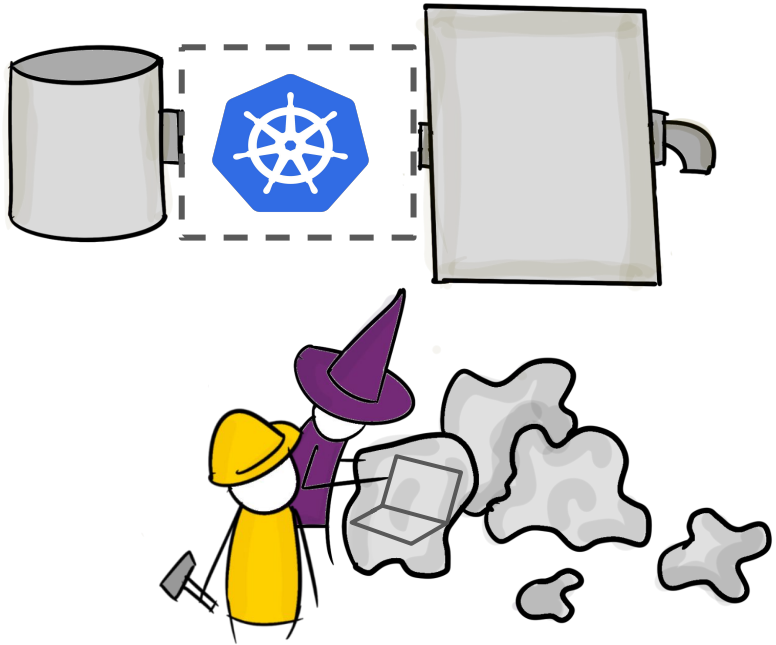




Illustrations courtesy of Ville Tuulos

Metaflow engineers paired with Data Scientists to experience their pains end-to-end





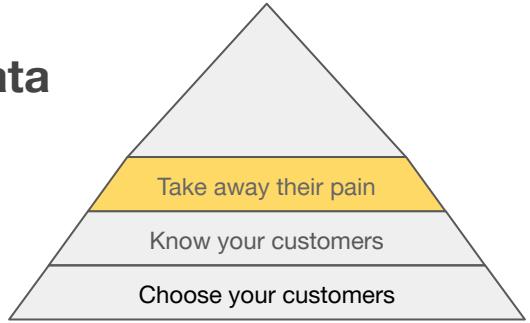
Illustrations courtesy of Ville Tuulos

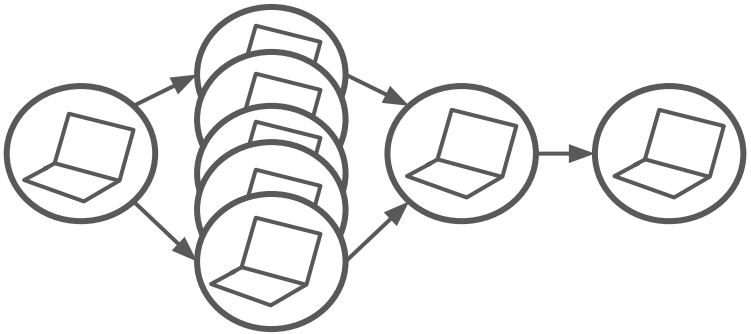
```
@step
def start(self):
  self.n_trees = [16, 32, 64]
  self.next(
    self.train, foreach='n_trees'
  )

@step
def train(self):
  n_trees = int(self.input)
  model, rmse = train_model(n_trees)
  self.rmse = rmse
  self.model = model
  self.next(self.join)

@step
def join(self, inputs):
  self.best_rmse = min(
    i.rmse for i in inputs
  )
  self.next(self.end)
```

The resulting features met Data Scientists where they were



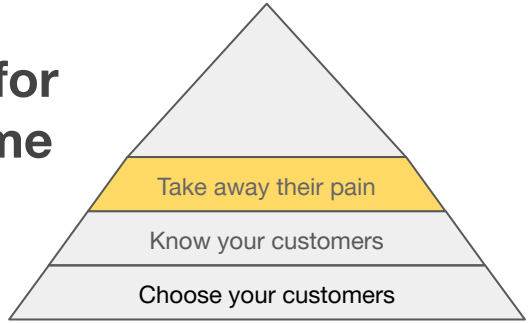


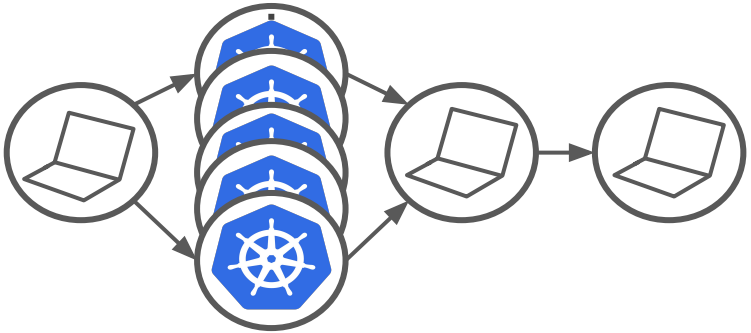
```
@step
def start(self):
  self.n_trees = [16, 32, 64]
  self.next(
    self.train, foreach='n_trees'
  )

@step
@kubernetes(memory=64000)
def train(self):
  n_trees = int(self.input)
  model, rmse = train_model(n_trees)
  self.rmse = rmse
  self.model = model
  self.next(self.join)

@step
def join(self, inputs):
  self.best_rmse = min(
    i.rmse for i in inputs
  )
  self.next(self.end)
```

Data Scientists could ask for compute resources any time they needed!



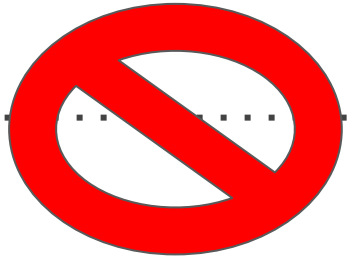


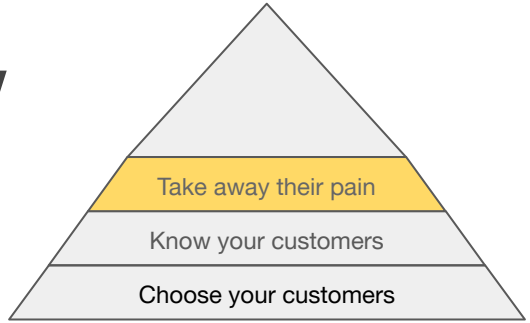
```python
@step
def start(self):
  self.n_trees = [16, 32, 64]
  self.next(
    self.train, foreach='n_trees'
  )

@step
@kubernetes(memory=64000)
def train(self):
  n_trees = int(self.input)
  model, rmse = train_model(n_trees)
  self.rmse = rmse
  self.model = model
  self.next(self.join)

@step
def join(self, inputs):
  self.best_rmse = min(
    i.rmse for i in inputs
  )
  self.next(self.end)
```

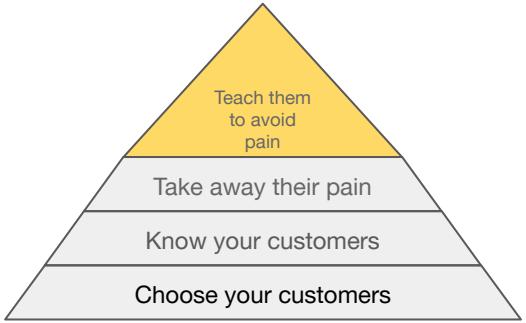
...Without introducing new pain

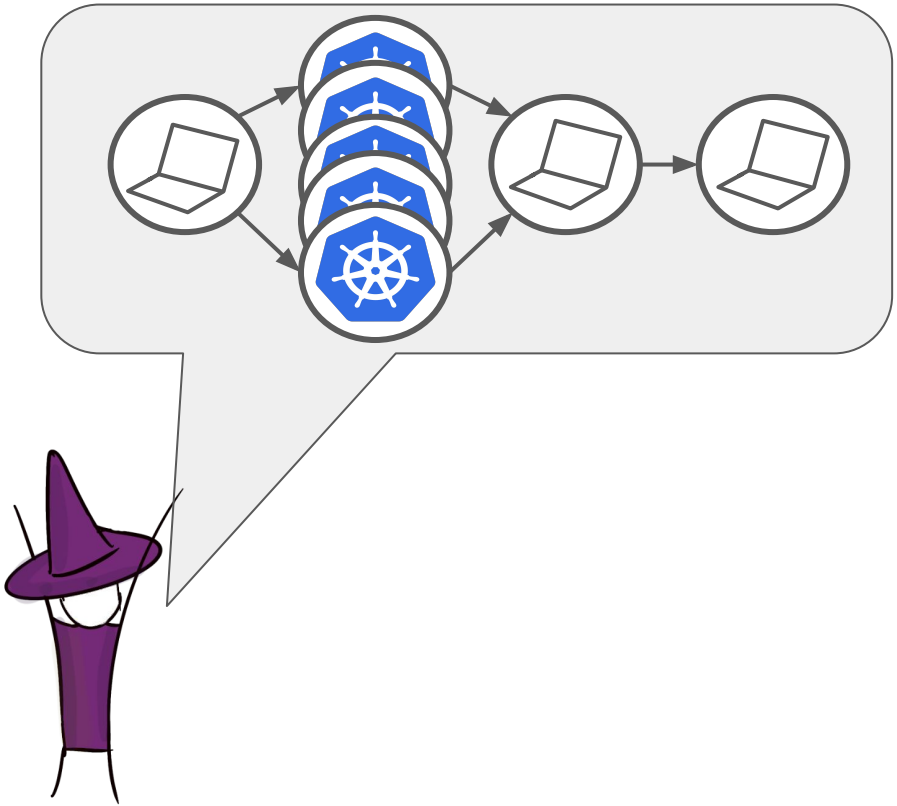





```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
      restartPolicy: Never
  backoffLimit: 4
```

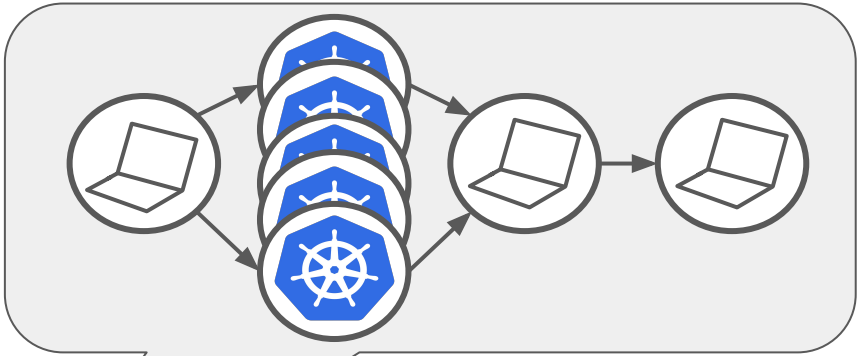
Data Scientists told their colleagues about the work they accomplished while using Metaflow

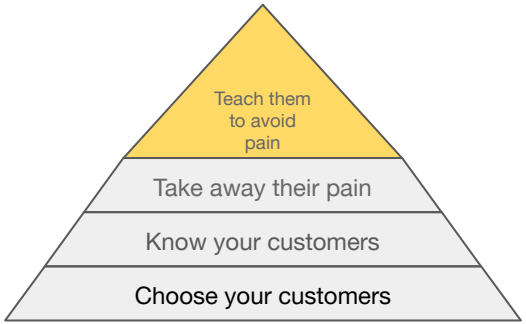




Illustrations courtesy of Ville Tuulos

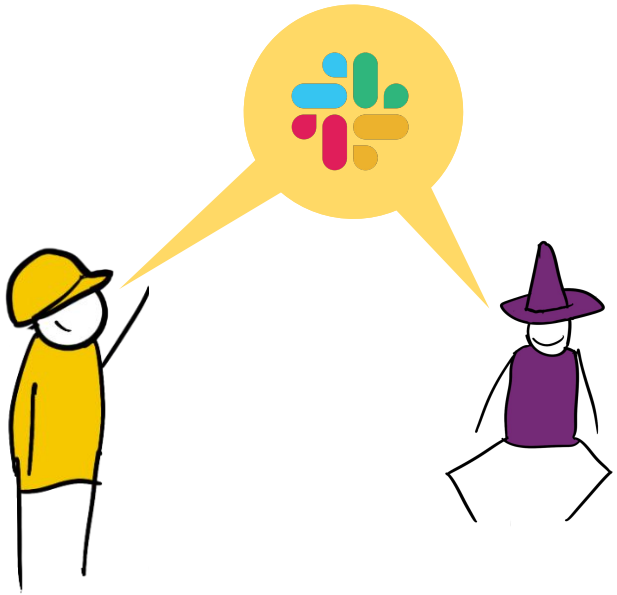
Metaflow engineers only marketed features that we could support forever!

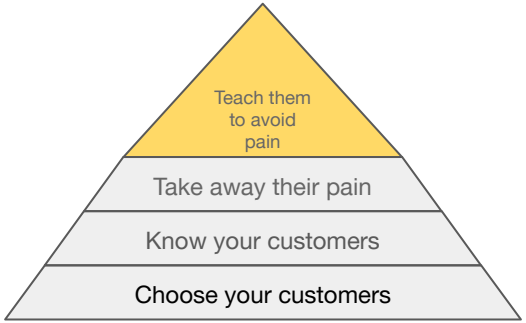


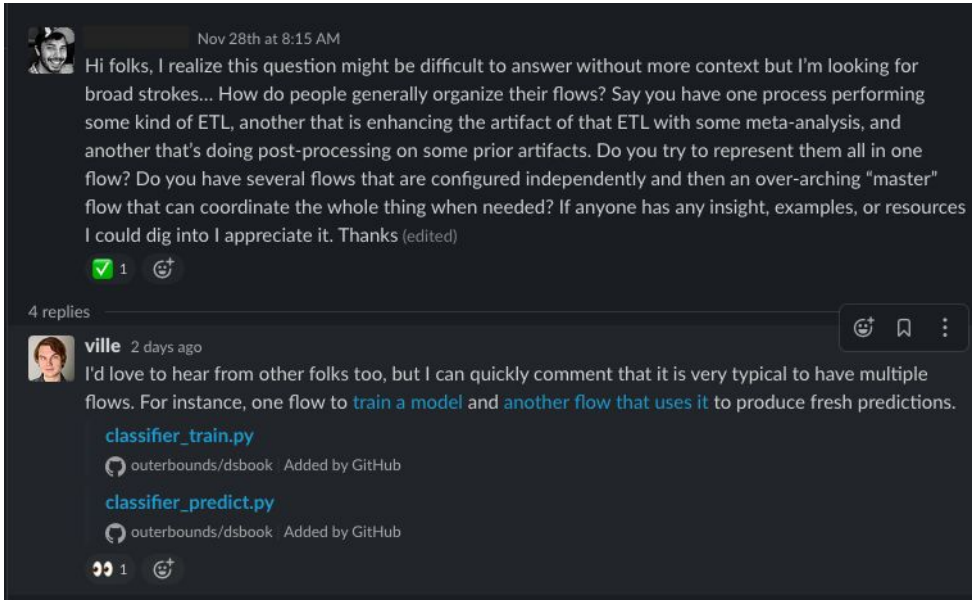


Illustrations courtesy of Ville Tuulos

The Metaflow team provided fanatical customer support

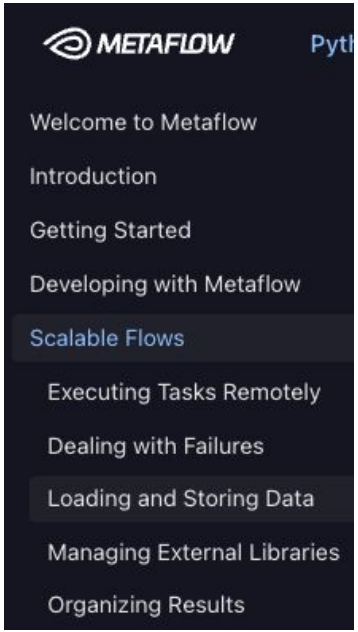


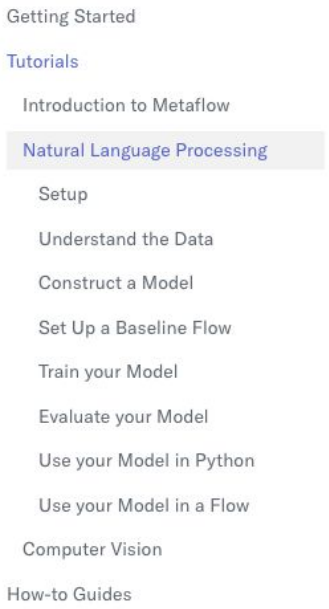


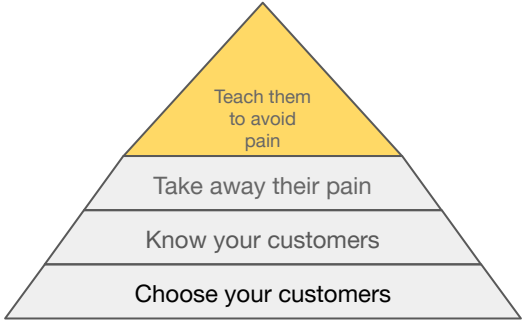


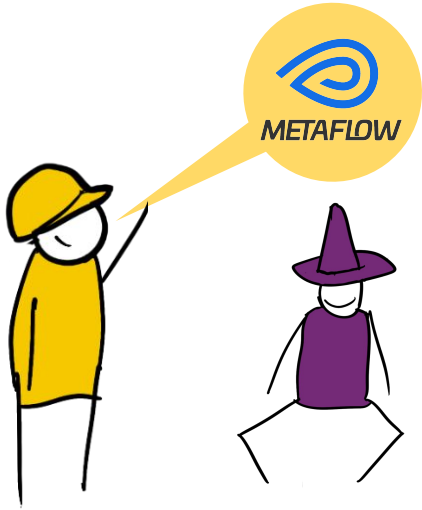
Further reading: [Rackspace on Fanatical Customer Service]

Metaflow engineers treated education as part of of the product





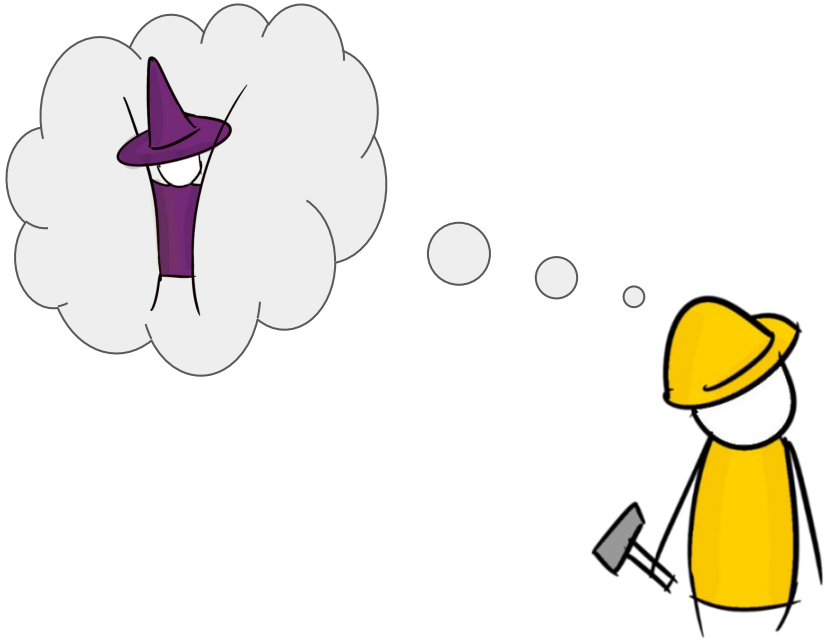




Documentation

Tutorials

Classes

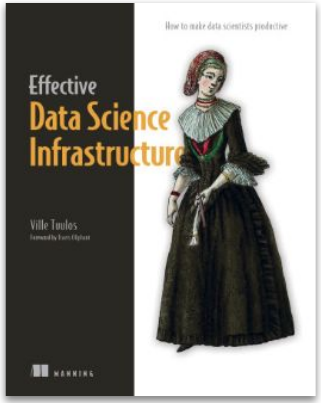


Illustrations courtesy of Ville Tuulos

Want to learn more about Metaflow?

docs.metaflow.org

outerbounds.com



Effective Data Science Infrastructure
New book, by Ville Tuulos

Big thanks to Ville, who let me use many of his slides and illustrations in this deck!

I’m looking for my next adventure!

Find me after the talk, or visit my LinkedIn:

linkedin.com/in/julieamundson

Questions?