Copyright Notice

These slides are distributed under the Creative Commons License.

DeepLearning.AI makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite DeepLearning.AI as the source of the slides.

For the rest of the details of the license, see
https://creativecommons.org/licenses/by-sa/2.0/legalcode

DeepLearning.AI

Collecting, Labeling, and Validating Data

Welcome

The importance of data

“Data is the hardest part of ML and the most important piece to get right... Broken data is the most common cause of problems in production ML systems”

- Scaling Machine Learning at Uber with Michelangelo - Uber

“No other activity in the machine learning life cycle has a higher return on investment than improving the data a model has access to.”

- Feast: Bridging ML Models and Data - Gojek

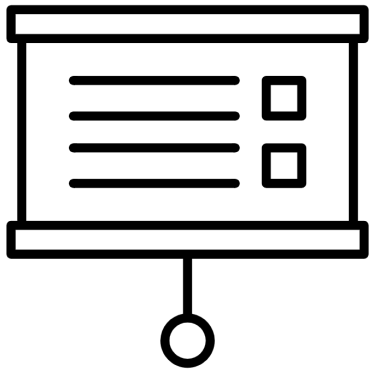
DeepLearning.AI

Introduction to Machine Learning Engineering for Production

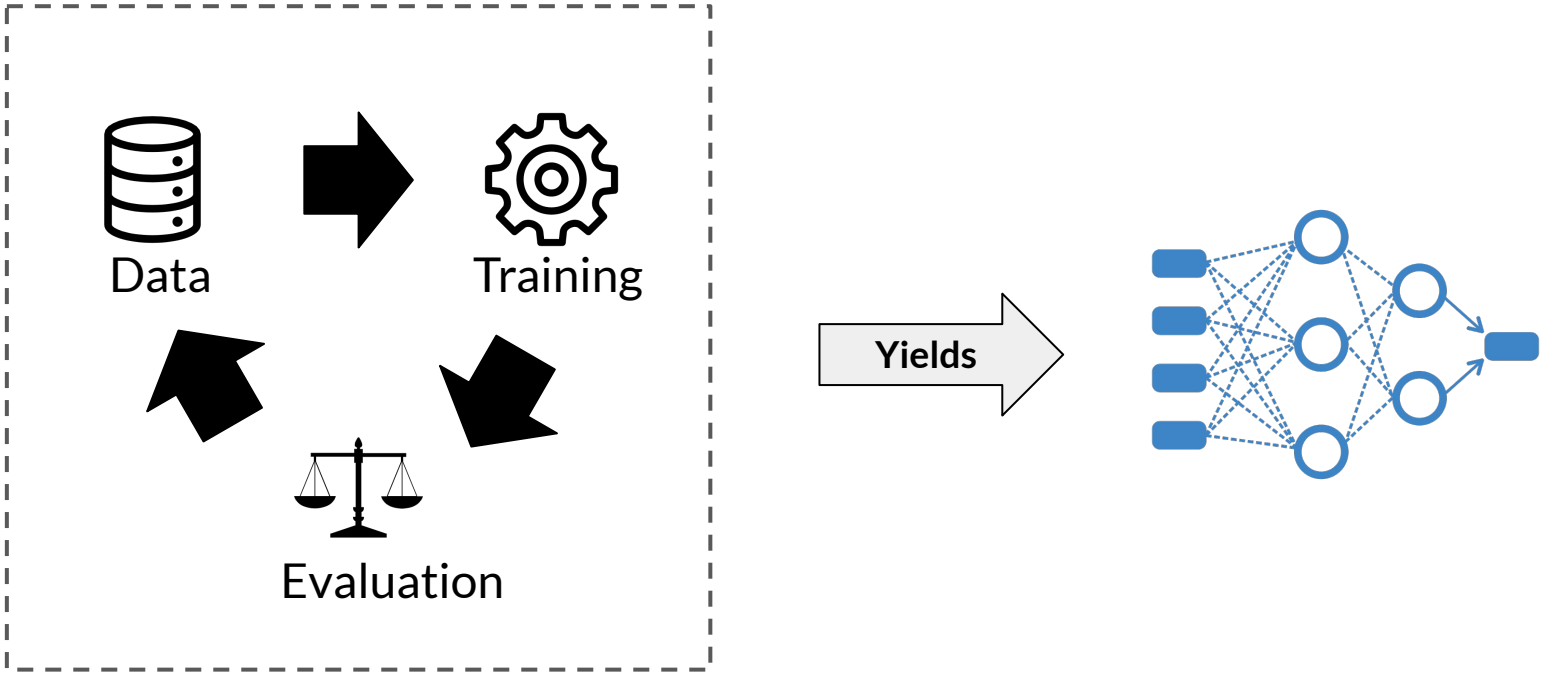
Overview

Outline

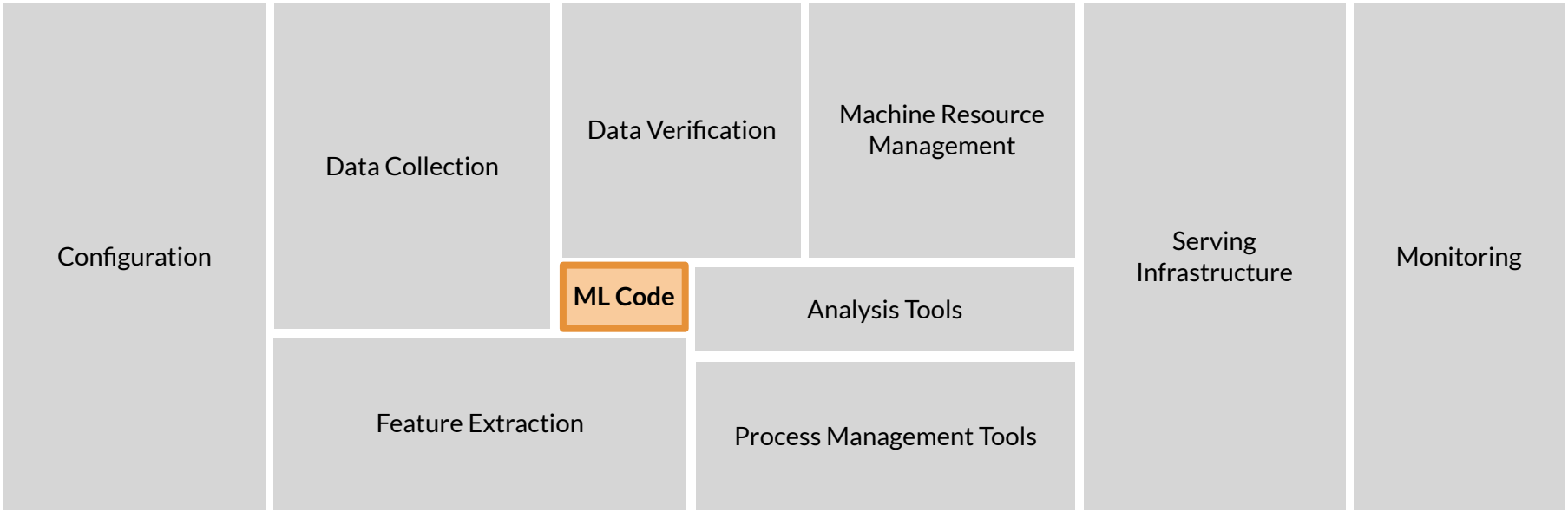
- Machine learning (ML) engineering for production: overview
- Production ML = ML development + software development
- Challenges in production ML



Traditional ML modeling



Production ML systems require so much more



ML modeling vs production ML

	Academic/Research ML	Production ML
Data	Static	Dynamic - Shifting
Priority for design	Highest overall accuracy	Fast inference, good interpretability
Model training	Optimal tuning and training	Continuously assess and retrain
Fairness	Very important	Crucial
Challenge	High accuracy algorithm	Entire system

Production machine learning

Machine learning development

+

Modern software development

Managing the entire life cycle of data

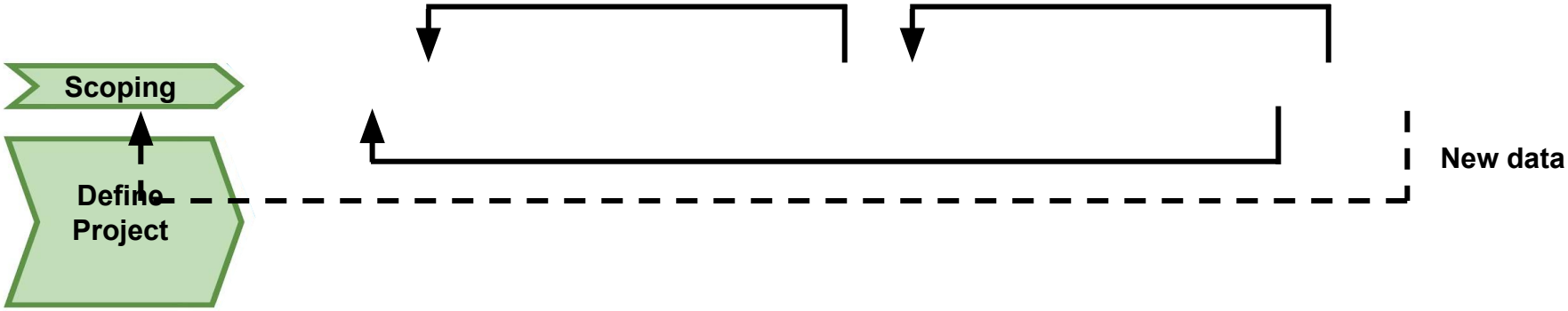
- Labeling
- Feature space coverage
- Minimal dimensionality
- Maximum predictive data
- Fairness
- Rare conditions

Modern software development

Accounts for:

- Scalability
- Extensibility
- Configuration
- Consistency & reproducibility
- Safety & security
- Modularity
- Testability
- Monitoring
- Best practices

Production machine learning system



Challenges in production grade ML

- Build integrated ML systems
- Continuously operate it in production
- Handle continuously changing data
- Optimize compute resource costs

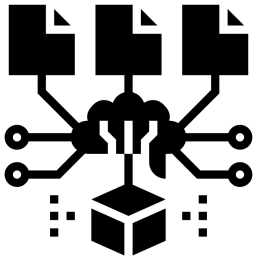
DeepLearning.AI

Introduction to Machine Learning Engineering for Production

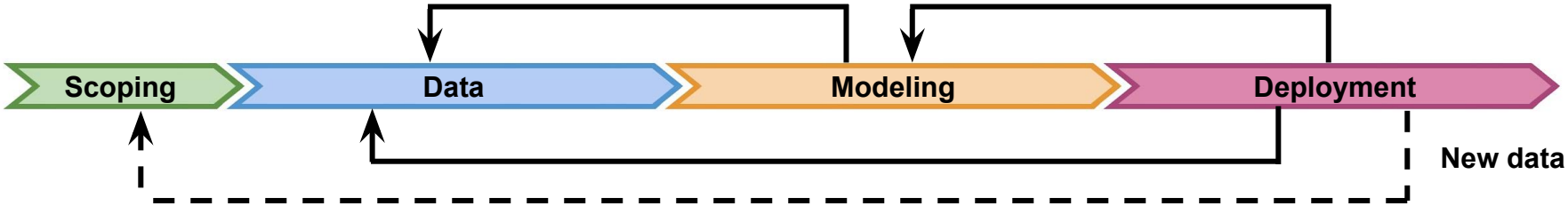
ML Pipelines

Outline

- ML Pipelines
- Directed Acyclic Graphs and Pipeline Orchestration Frameworks
- Intro to TensorFlow Extended (TFX)



ML pipelines



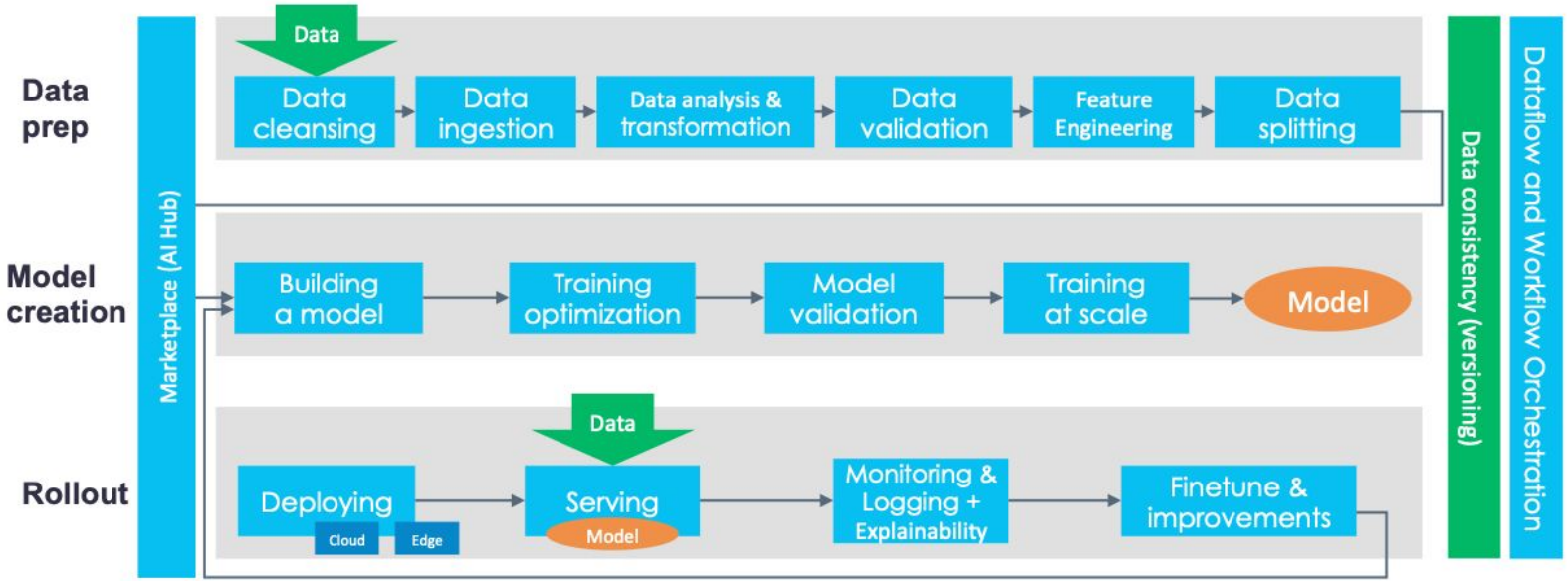
Infrastructure for

automating, monitoring, and maintaining

model training and deployment

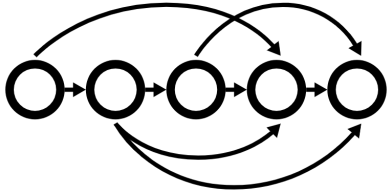
Production ML infrastructure

CD Foundation MLOps reference architecture

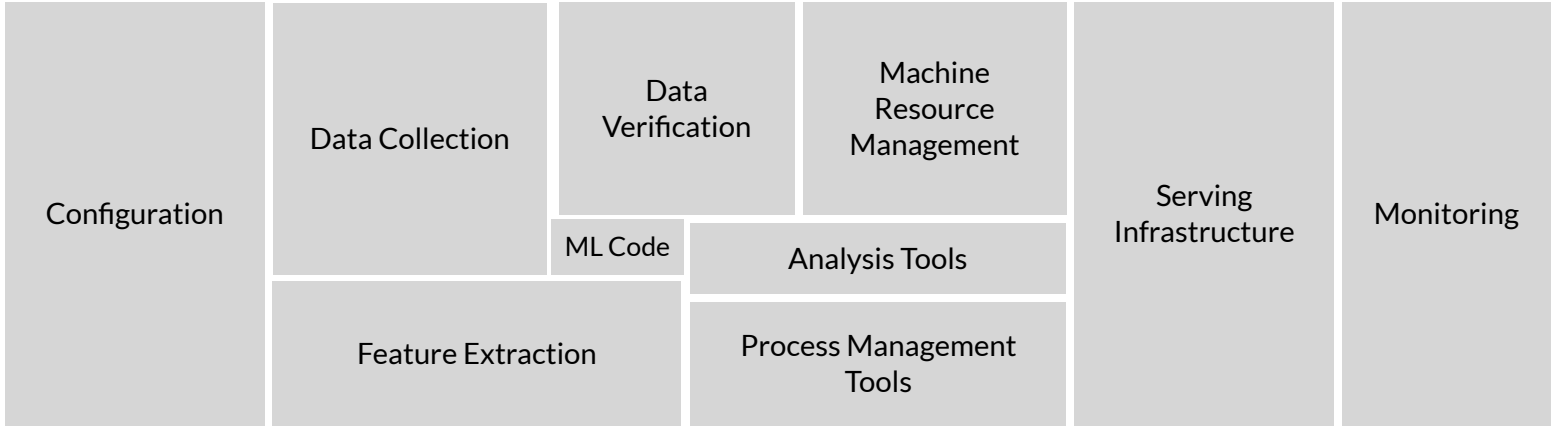


Directed acyclic graphs

- A directed acyclic graph (DAG) is a directed graph that has no cycles
- ML pipeline workflows are usually DAGs
- DAGs define the sequencing of the tasks to be performed, based on their relationships and dependencies.



Pipeline orchestration frameworks



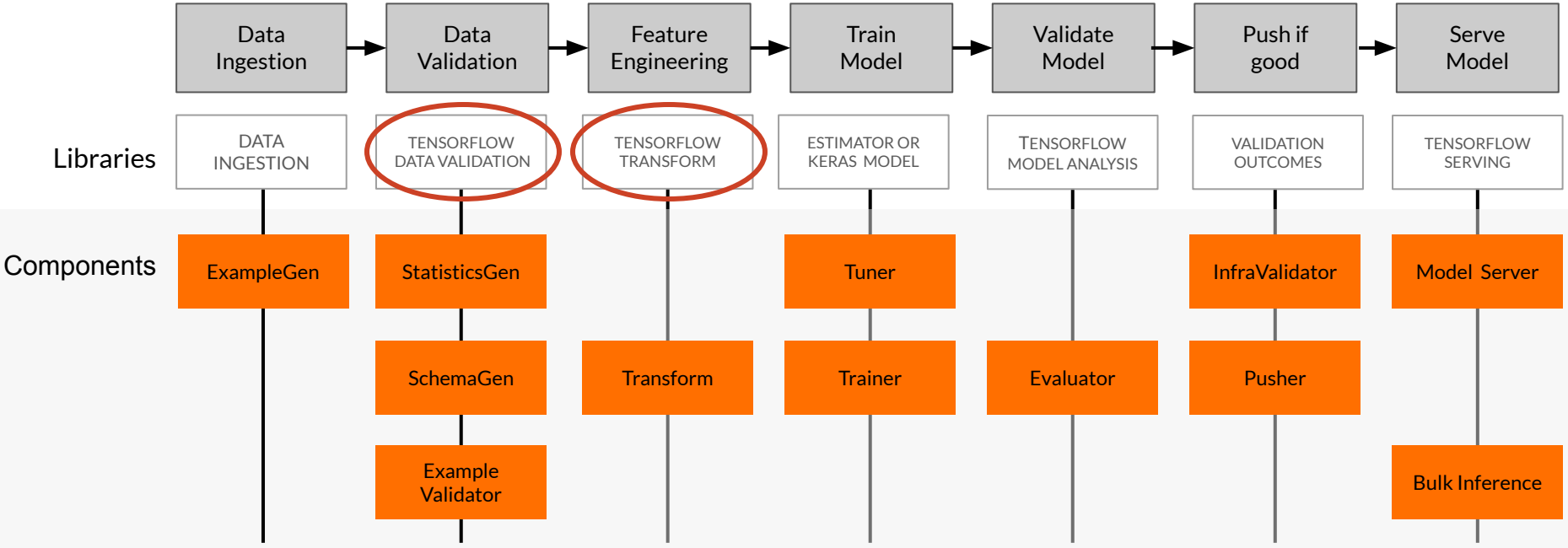
- Responsible for scheduling the various components in an ML pipeline DAG dependencies
- Help with pipeline automation
- Examples: Airflow, Argo, Celery, Luigi, Kubeflow

TensorFlow Extended (TFX)

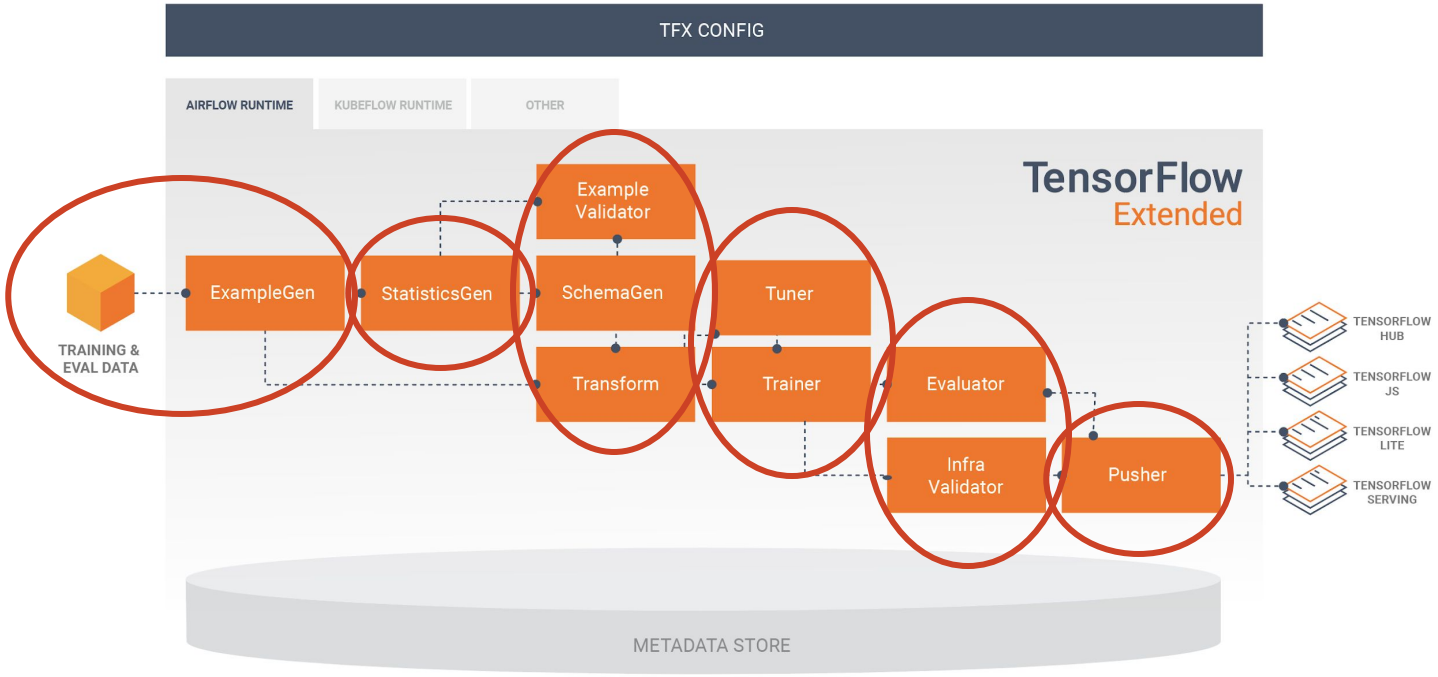
End-to-end platform for deploying production ML pipelines

Sequence of components that are designed for scalable, high-performance machine learning tasks

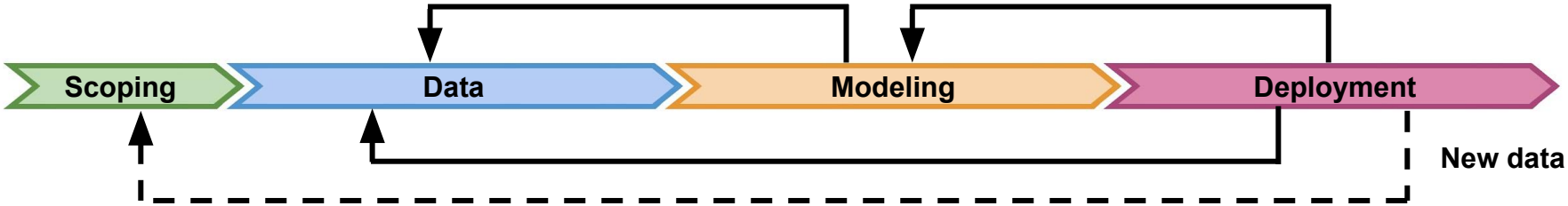
TFX production components



TFX Hello World



Key points

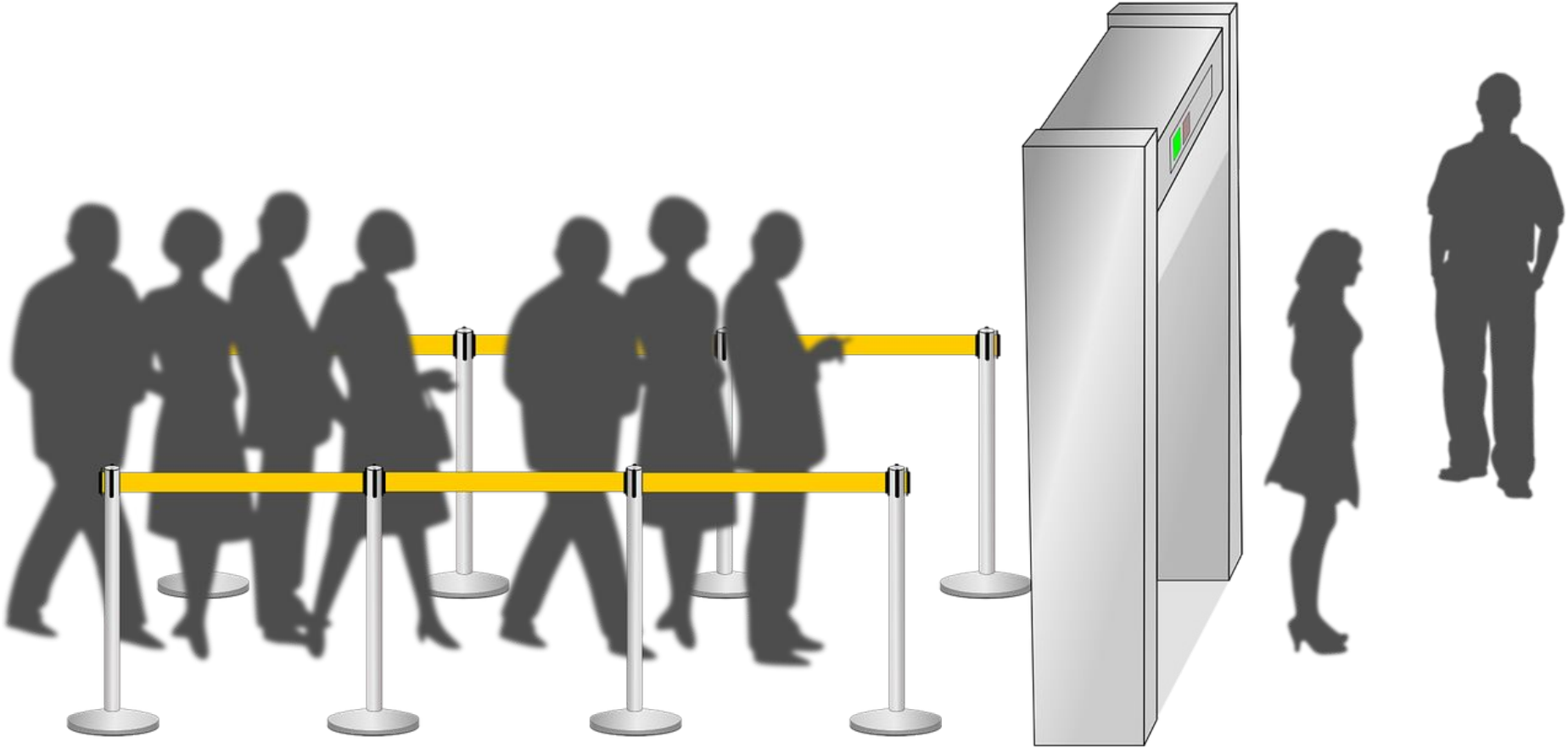


- Production ML pipelines: automating, monitoring, and maintaining end-to-end processes
- Production ML is much more than just ML code
    - ML development + software development
- TFX is an open-source end-to-end ML platform

DeepLearning.AI

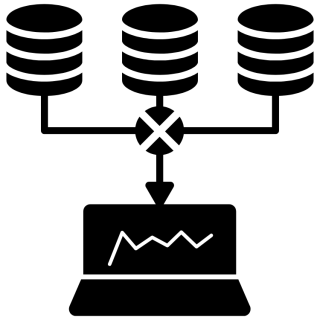
Collecting Data

Importance of Data



Outline

- Importance of data quality
- Data pipeline: data collection, ingestion and preparation
- Data collection and monitoring



The importance of data

"Data is the hardest part of ML and the most important piece to get right... Broken data is the most common cause of problems in production ML systems"

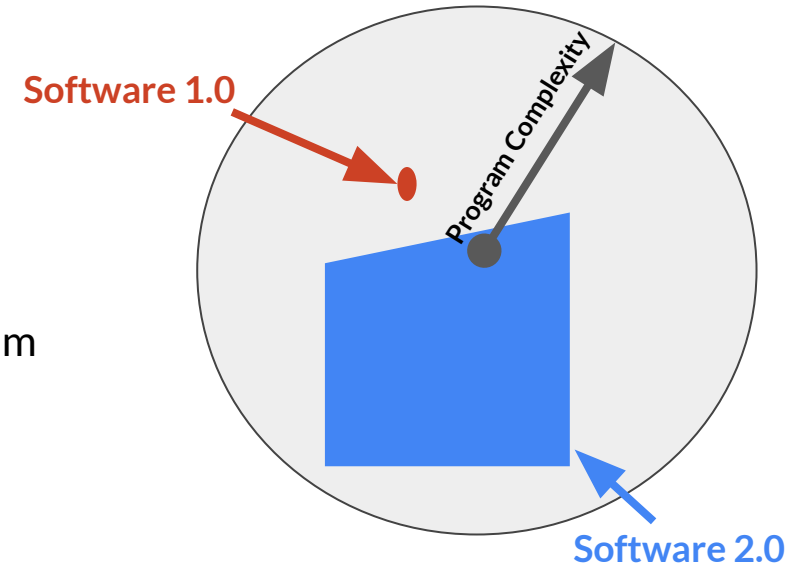
- Scaling Machine Learning at Uber with Michelangelo - Uber

"No other activity in the machine learning life cycle has a higher return on investment than improving the data a model has access to."

- Feast: Bridging ML Models and Data - Gojek

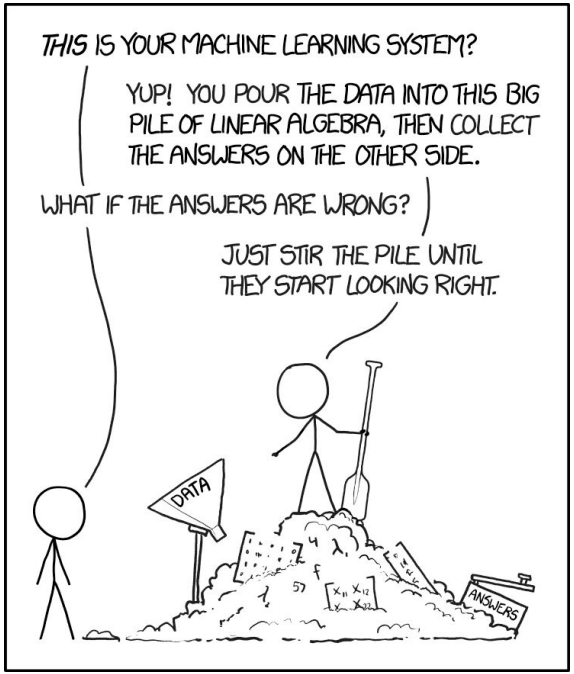
ML: Data is a first class citizen

- Software 1.0
  - Explicit instructions to the computer
- Software 2.0
  - Specify some goal on the behavior of a program
  - Find solution using optimization techniques
  - Good data is key for success
  - Code in Software = Data in ML

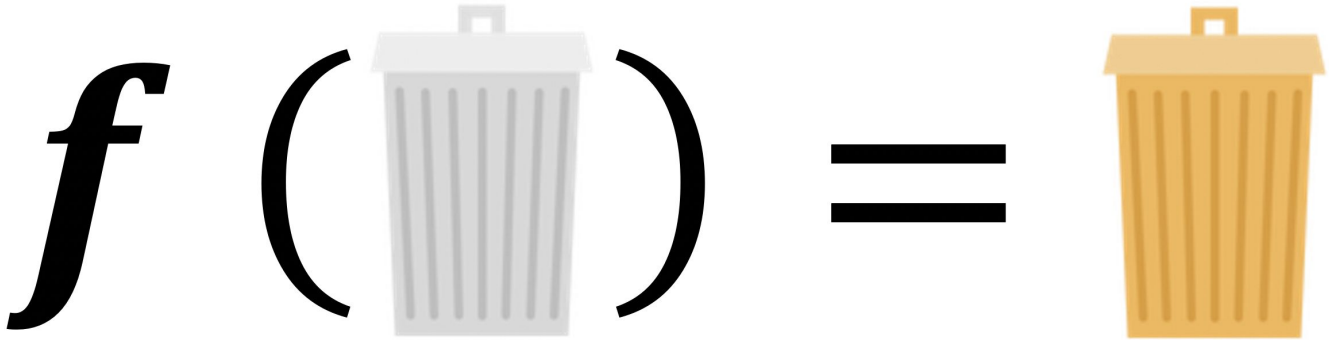


Everything starts with data

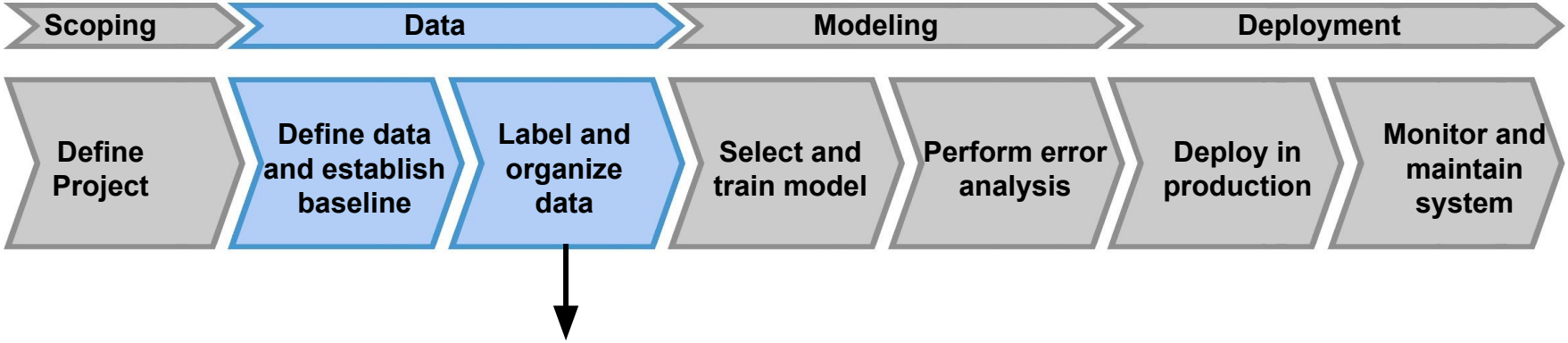
- Models aren't magic
- Meaningful data:
  - maximize predictive content
  - remove non-informative data
  - feature space coverage



Garbage in, garbage out

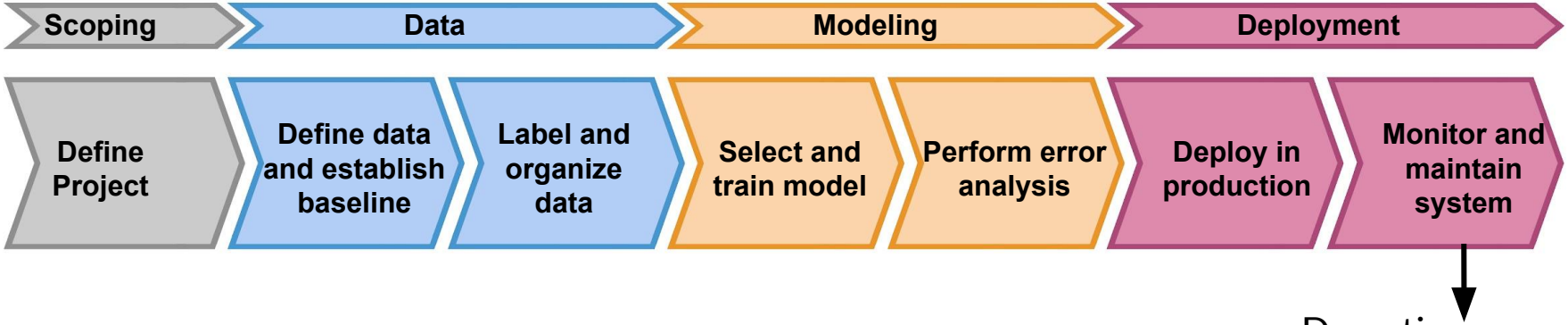


Data pipeline



- Data collection
- Data ingestion
- Data formatting
- Feature engineering
- Feature extraction

Data collection and monitoring



- Downtime
- Errors
- Distributions shifts
- Data failure
- Service failure

Key Points

- Understand users, translate user needs into data problems
- Ensure data coverage and high predictive signal
- Source, store and monitor quality data responsibly

DeepLearning.AI

Collecting Data

Example Application: Suggesting Runs

 Example application: Suggesting runs

Users	Runners
User Need	Run more often
User Actions	Complete run using the app
ML System Output	• What routes to suggest • When to suggest them
ML System Learning	• Patterns of behaviour around accepting run prompts • Completing runs • Improving consistency

Key considerations

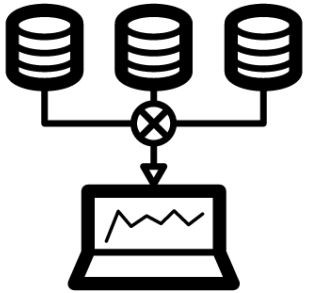
- Data availability and collection
  - What kind of/how much data is available?
  - How often does the new data come in?
  - Is it annotated?
    - If not, how hard/expensive is it to get it labeled?
- Translate user needs into data needs
  - Data needed
  - Features needed
  - Labels needed

Example dataset

	FEATURES				
	Runner ID	**Run**	**Runner Time**	**Elevation**	**Fun**
EXAMPLES	AV3DE	Boston Marathon	03:40:32	1,300 ft	Low
	X8KGF	Seattle Oktoberfest 5k	00:35:40	0 ft	High
	BH9IU	Houston Half-marathon	02:01:18	200 ft	Medium
					LABELS

Get to know your data

- Identify data sources
- Check if they are refreshed
- Consistency for values, units, & data types
- Monitor outliers and errors



Dataset issues

- Inconsistent formatting
  - Is zero “0”, “0.0”, or an indicator of a missing measurement
- Compounding errors from other ML Models
- Monitor data sources for system issues and outages

Measure data effectiveness

- Intuition about data value can be misleading
  - Which features have predictive value and which ones do not?
- Feature engineering helps to maximize the predictive signals
- Feature selection helps to measure the predictive signals

Translate user needs into data needs

Data Needed	• Running data from the app • Demographic data • Local geographic data

Translate user needs into data needs

Features Needed	• Runner demographics • Time of day • Run completion rate • Pace • Distance ran • Elevation gained • Heart rate

Translate user needs into data needs

Labels Needed	• Runner acceptance or rejection of app suggestions • User generated feedback regarding why suggestion was rejected • User rating of enjoyment of recommended runs

Key points

- Understand your user, translate their needs into data problems
  - What kind of/how much data is available
  - What are the details and issues of your data
  - What are your predictive features
  - What are the labels you are tracking
  - What are your metrics

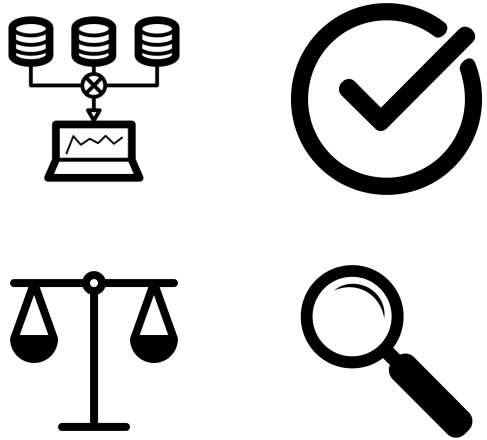
DeepLearning.AI

Collecting Data

Responsible Data: Security, Privacy & Fairness

Outline

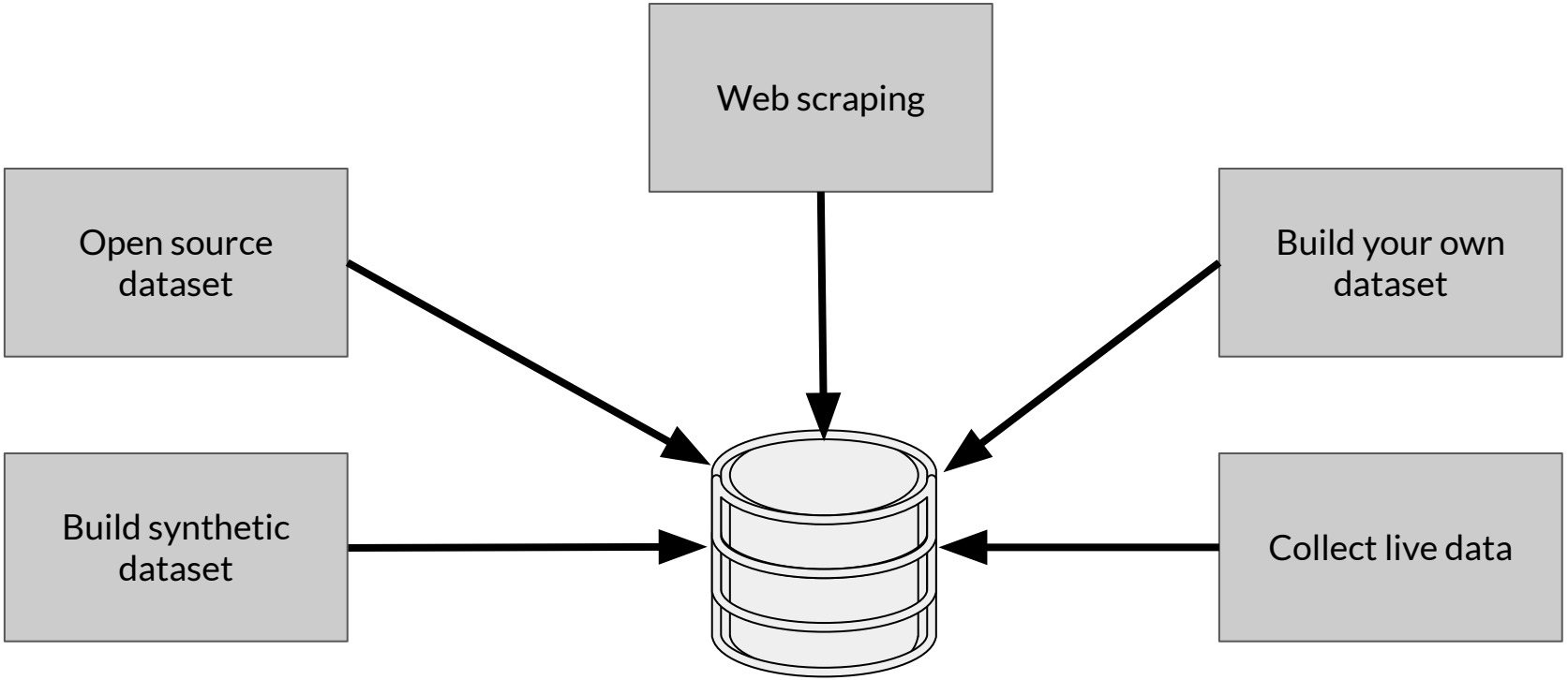
- Data Sourcing
- Data Security and User Privacy
- Bias and Fairness



Avoiding problematic biases in datasets

Example: classifier trained on the Open Images dataset

Source Data Responsibly



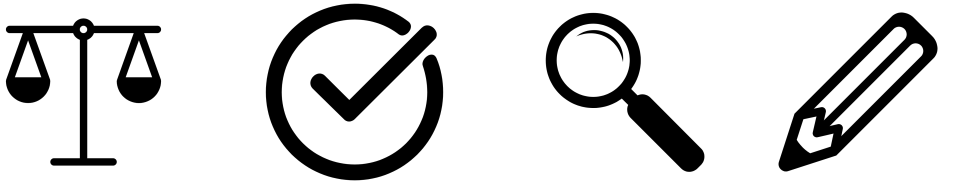
Data security and privacy

- Data collection and management isn’t just about your model
  - Give user control of what data can be collected
  - Is there a risk of inadvertently revealing user data?
- Compliance with regulations and policies (e.g. GDPR)

Users privacy

- Protect personally identifiable information
  - Aggregation - replace unique values with summary value
  - Redaction - remove some data to create less complete picture

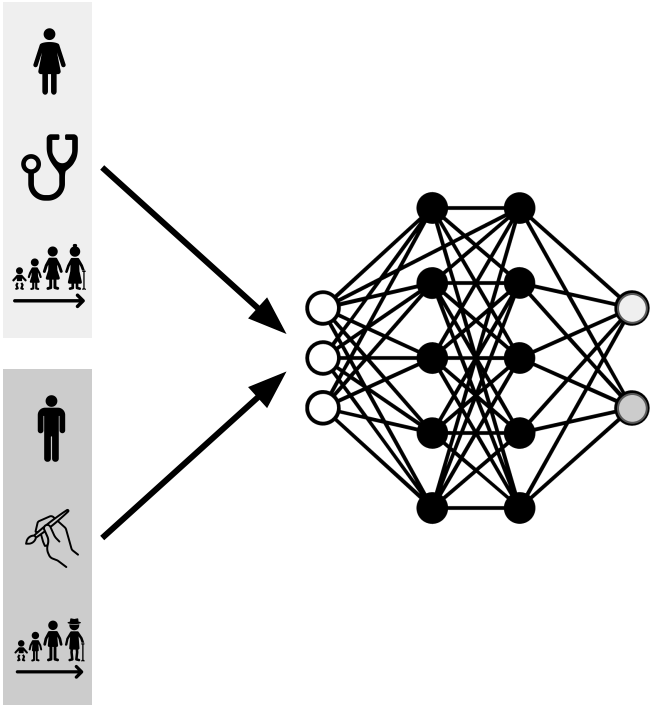
How ML systems can fail users



Fair Accountable Transparent Explainable

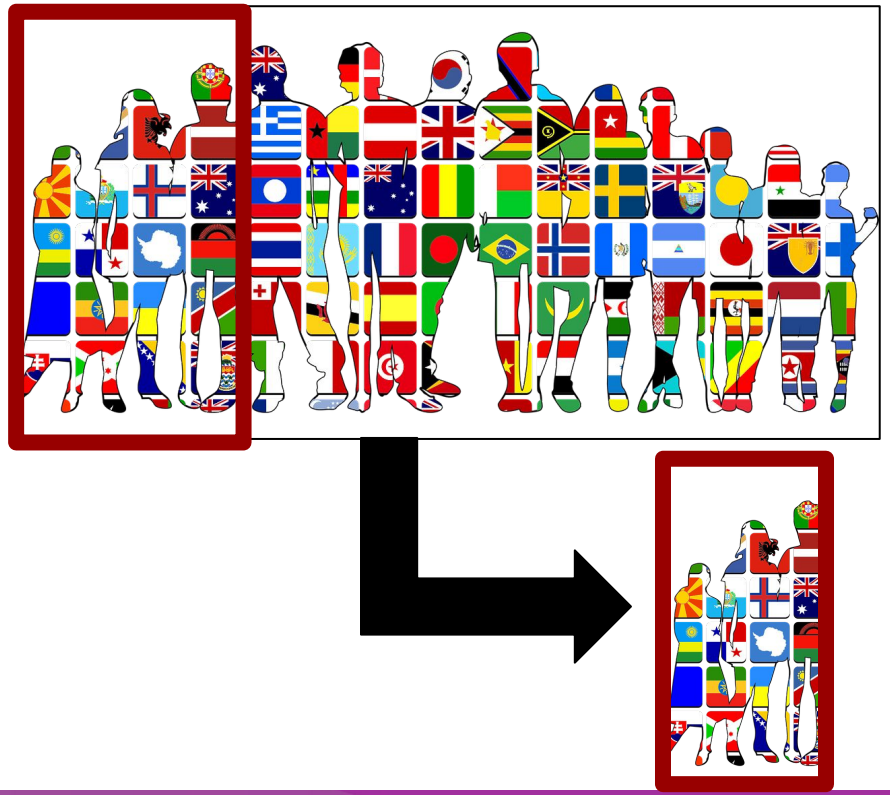
- Representational harm
- Opportunity denial
- Disproportionate product failure
- Harm by disadvantage

Commit to fairness



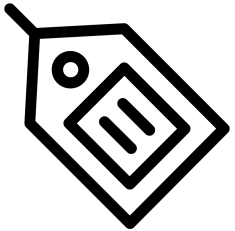
- Make sure your models are fair
  - Group fairness, equal accuracy
- Bias in human labeled and/or collected data
- ML Models can amplify biases

Biased data representation



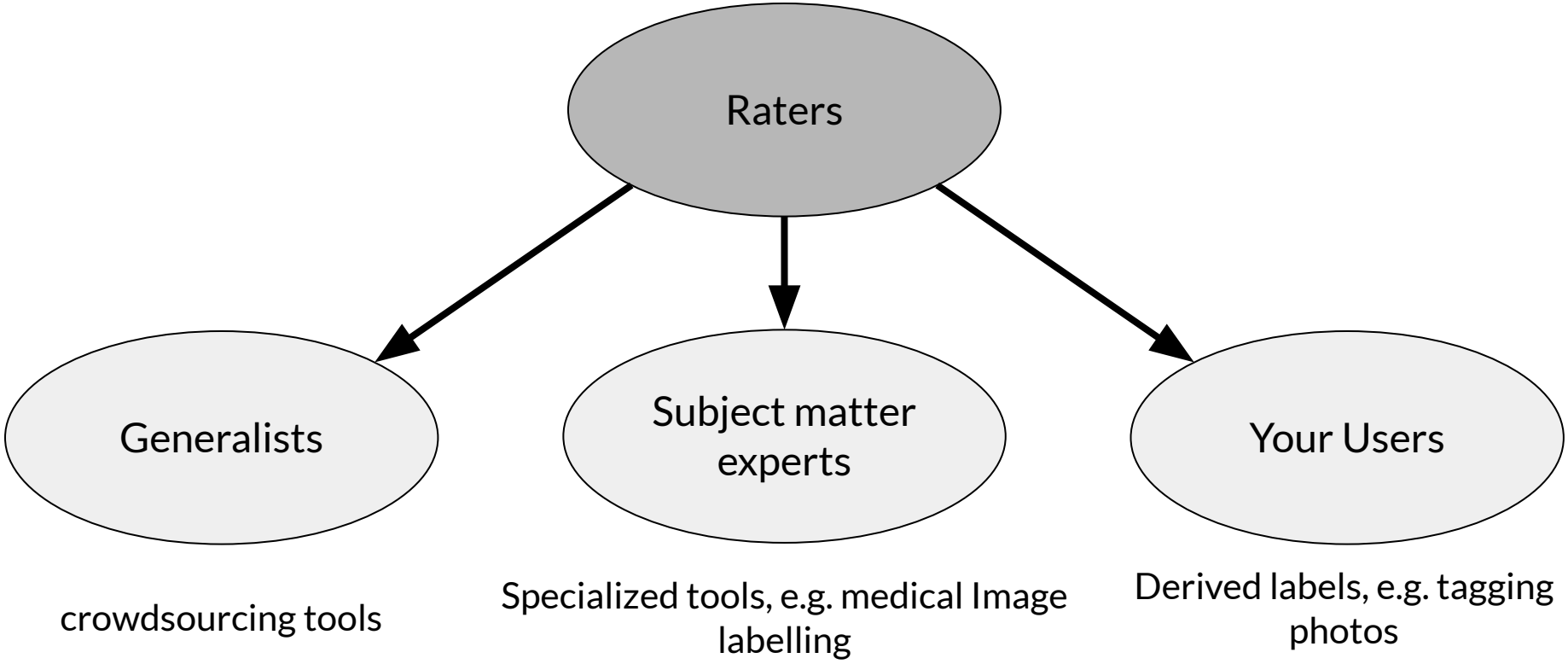
Reducing bias: Design fair labeling systems

- Accurate labels are necessary for supervised learning



- Labeling can be done by:
  - Automation (logging or weak supervision)
  - Humans (aka “Raters”, often semi-supervised)

Types of human raters



Key points

- Ensure rater pool diversity
- Investigate rater context and incentives
- Evaluate rater tools
- Manage cost
- Determine freshness requirements