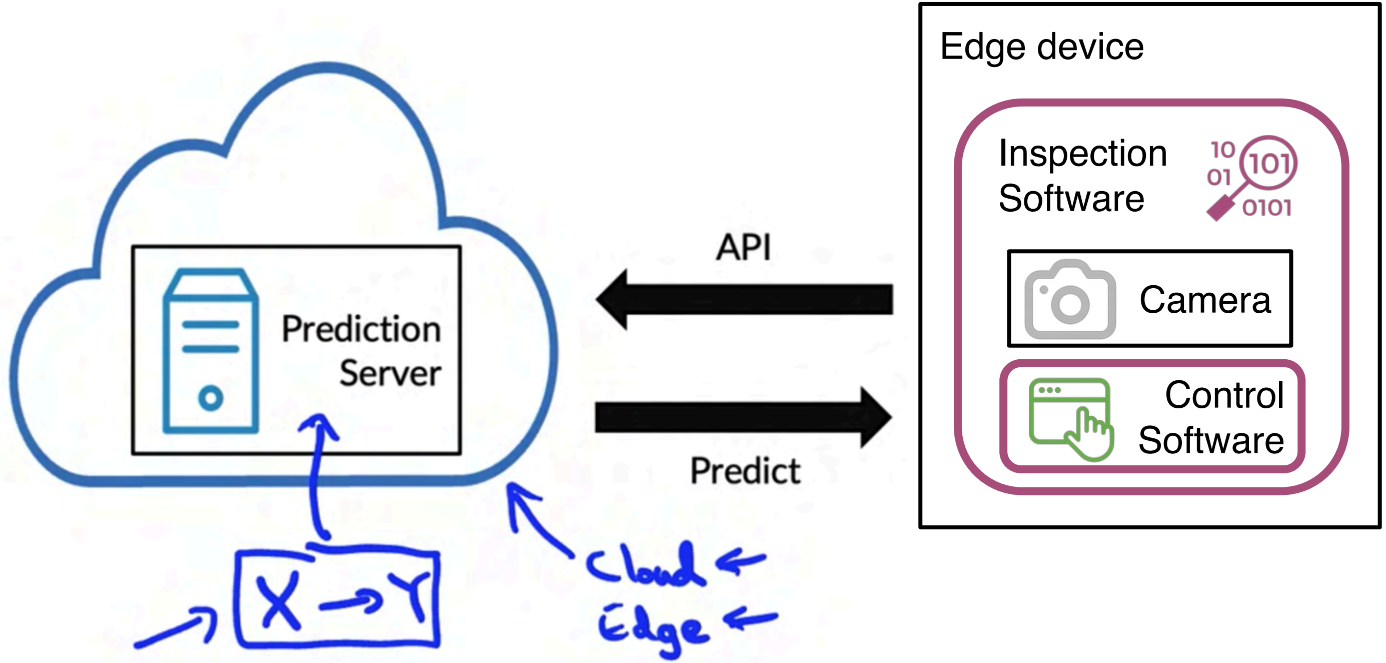
DeepLearning.AI

The Machine Learning Project Lifecycle

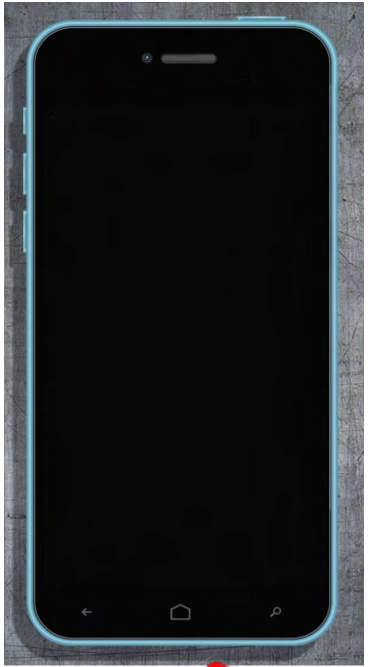
Welcome

Deployment example

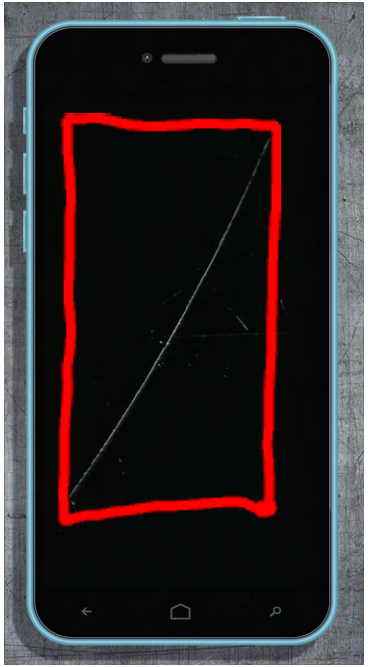
Photo from camera

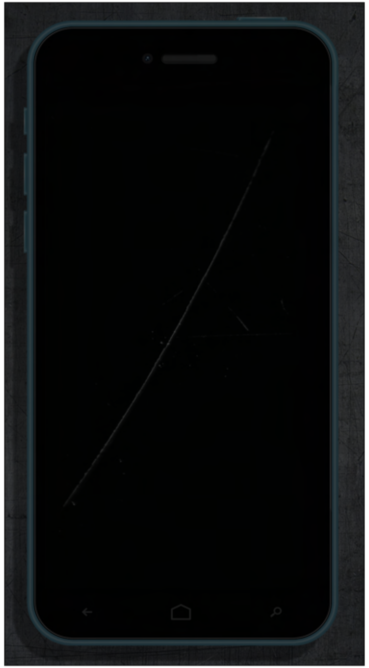


Visual inspection example

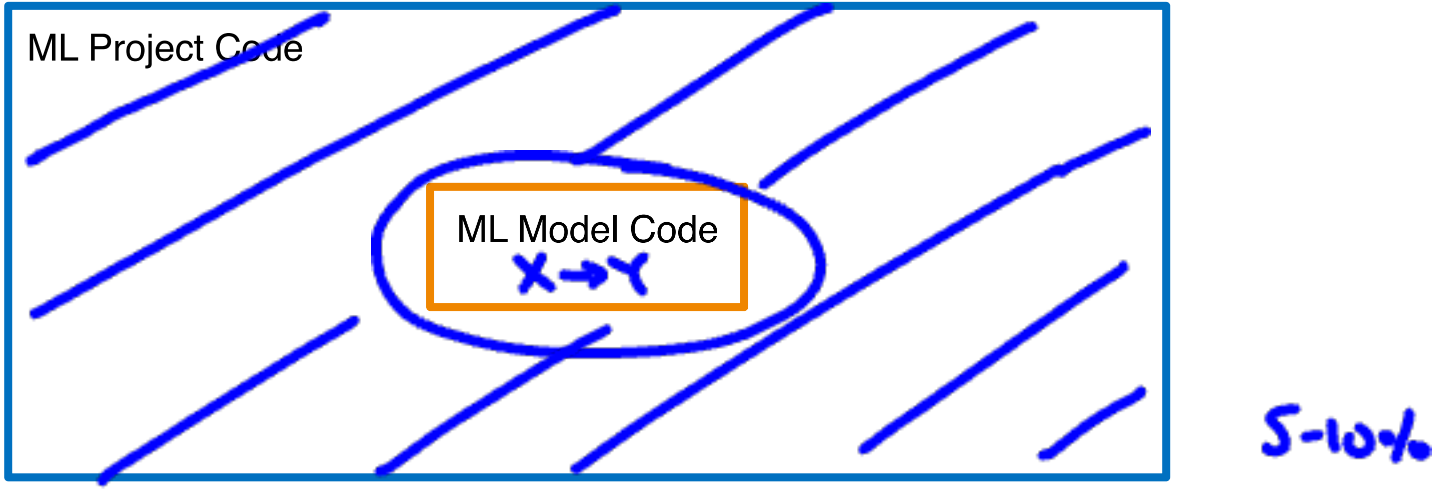


OK



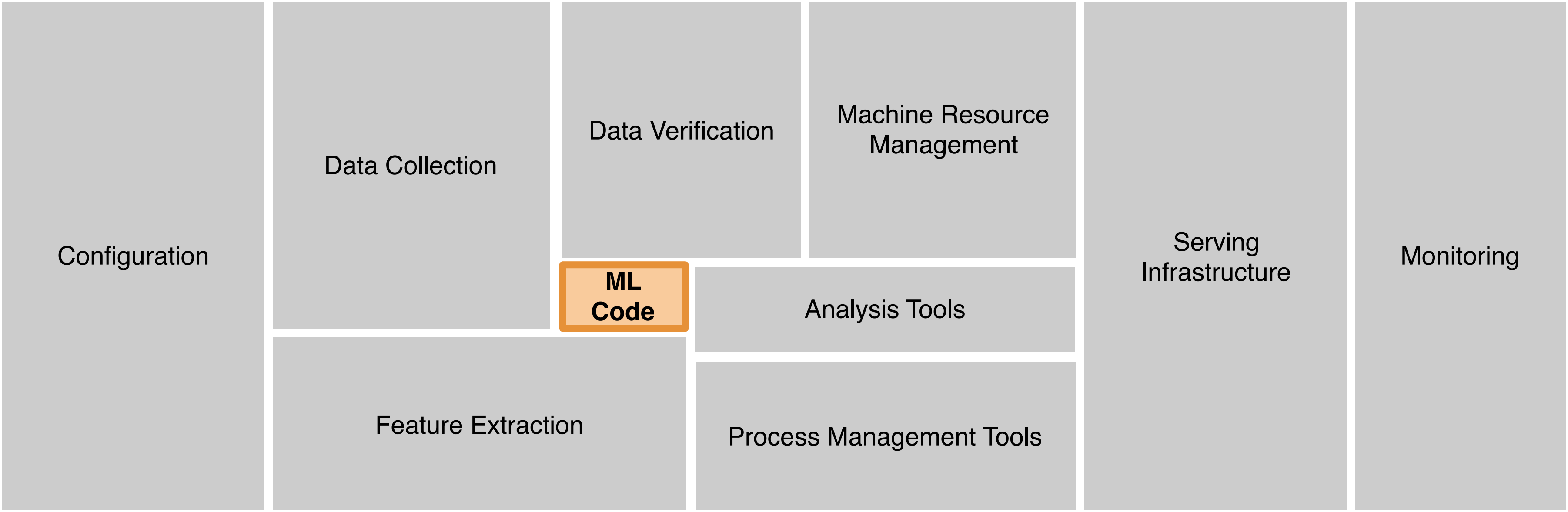


ML in production



"POC to Production Gap"

The requirements surrounding ML infrastructure



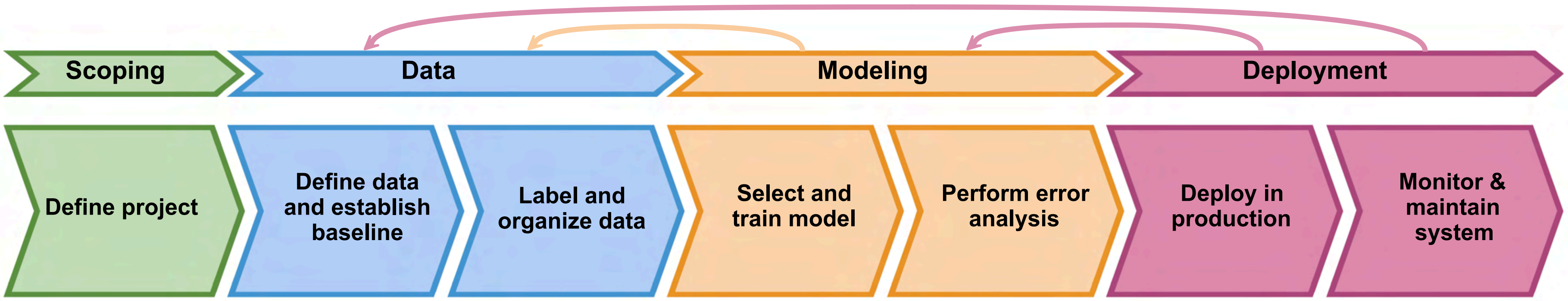
[D. Sculley et. al. NIPS 2015: Hidden Technical Debt in Machine Learning Systems]

DeepLearning.AI

The Machine Learning Project Lifecycle

Steps of an ML project

The ML project lifecycle



Scoping

Data

Modeling

Deployment

Define project

Define data and establish baseline

Label and organize data

Select and train model

Perform error analysis

Deploy in production

Monitor & maintain system

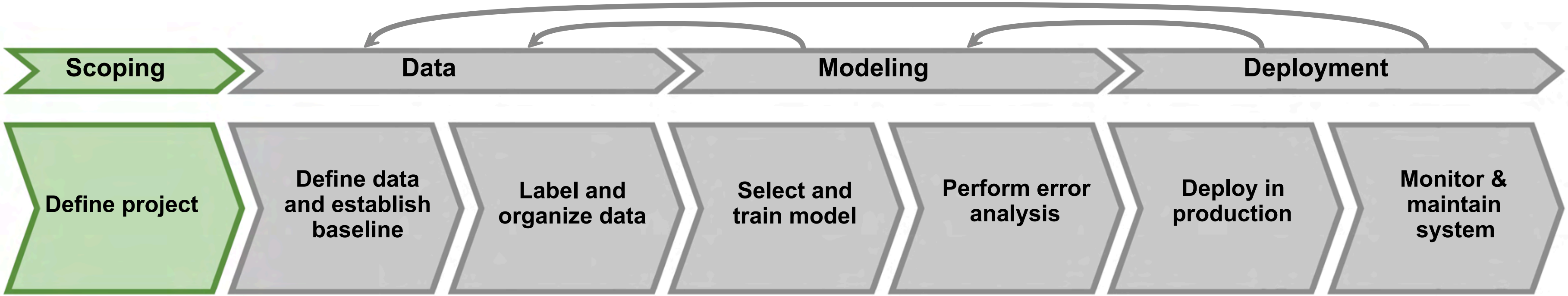
X → Y

DeepLearning.AI

The Machine Learning Project Lifecycle

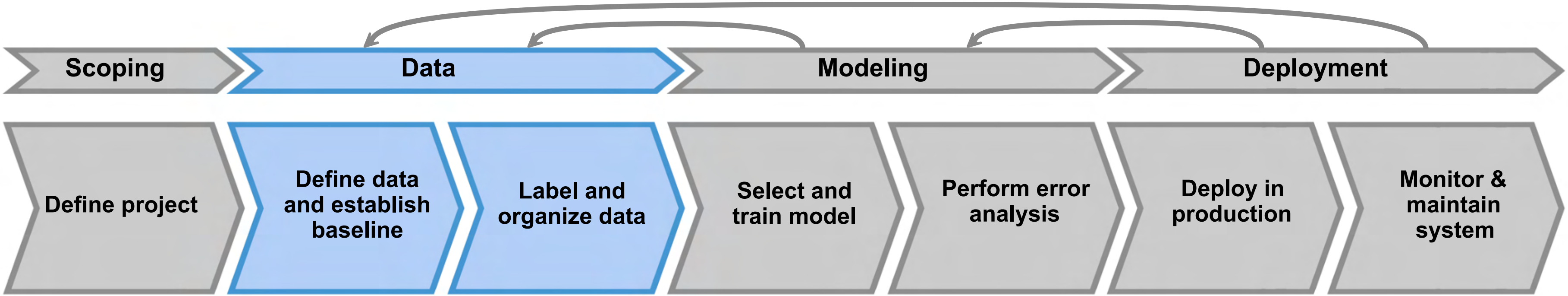
Case study:
speech recognition

Speech recognition: Scoping stage



- Decide to work on speech recognition for voice search.
- Decide on key metrics:
  - Accuracy, latency, throughput
- Estimate resources and timeline

Speech recognition: Data stage



Scoping

Data

Modeling

Deployment

Define project

Define data and establish baseline

Label and organize data

Select and train model

Perform error analysis

Deploy in production

Monitor & maintain system

Define data

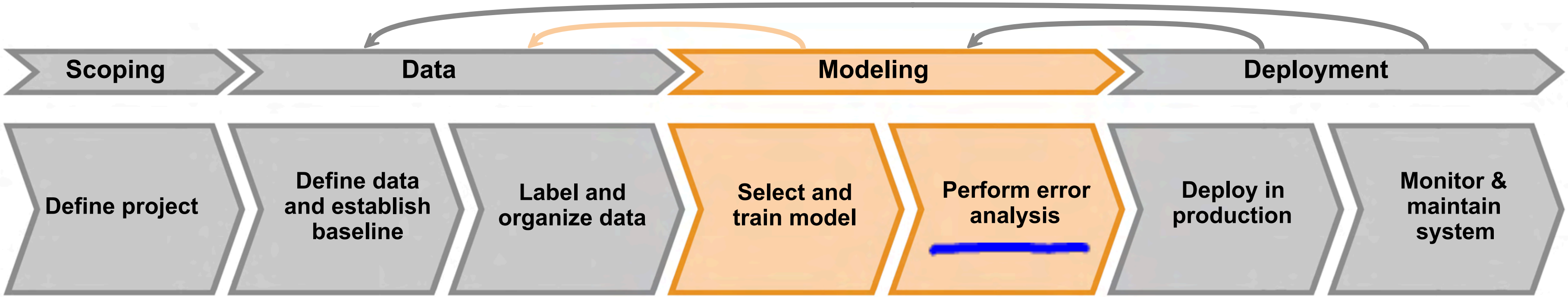
- Is the data labeled consistently?
- How much silence before/after each clip? 100ms 300ms 500ms
- How to perform volume normalization?

“Um, today’s weather” ✓

“Um… today’s weather”

“Today’s weather”

Speech recognition: Modeling stage



Scoping

Data

Modeling

Deployment

Define project

Define data and establish baseline

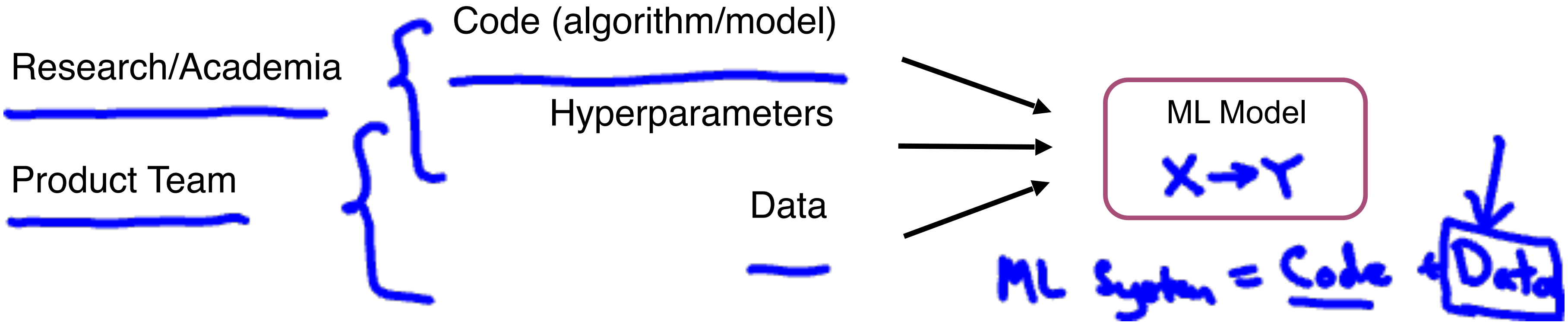
Label and organize data

Select and train model

Perform error analysis

Deploy in production

Monitor & maintain system



Research/Academia

Product Team

Code (algorithm/model)

Hyperparameters

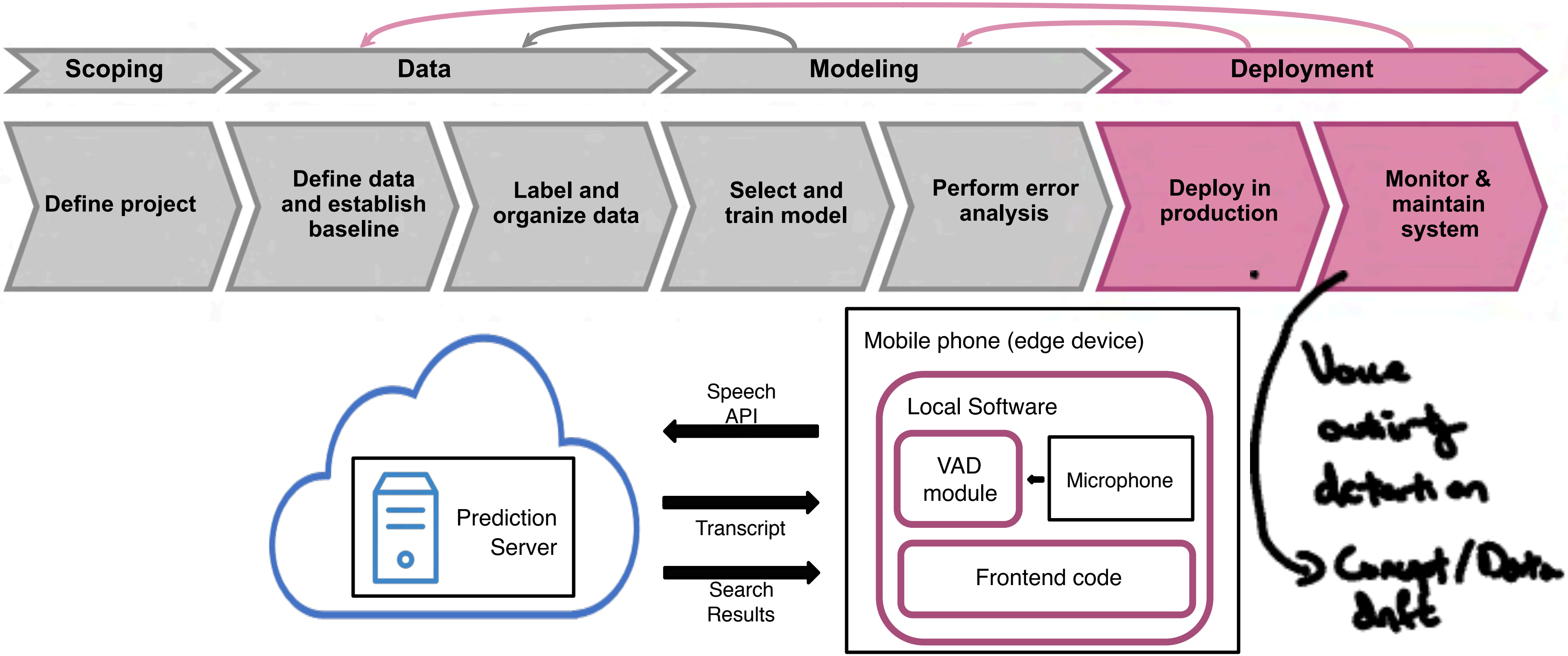
Data

ML Model

X→Y

ML System = Code + Data

Speech recognition: Deployment stage



Voice activity detection

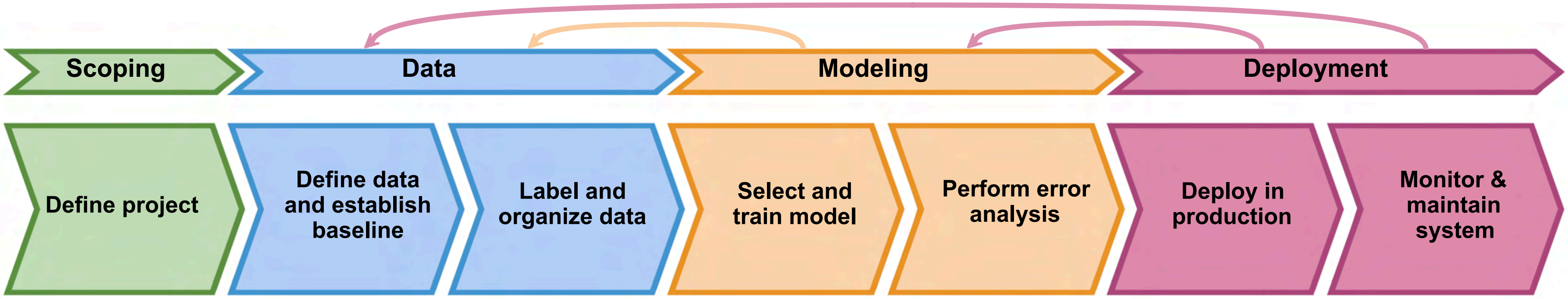
→ Concept/Data drift

DeepLearning.AI

The Machine Learning Project Lifecycle

Course outline

Course outline



1. Deployment
2. Modeling
3. Data

Optional: Scoping

MLOps (Machine Learning Operations) is an emerging discipline, and comprises a set of tools and principles to support progress through the ML project lifecycle.

DeepLearning.AI

Deployment

Key challenges

Concept drift and Data drift

$x \rightarrow y$

x

Speech recognition example

Training set: $x \rightarrow y$

- Purchased data, historical user data with transcripts

Test set:

- Data from a few months ago

Gradual change

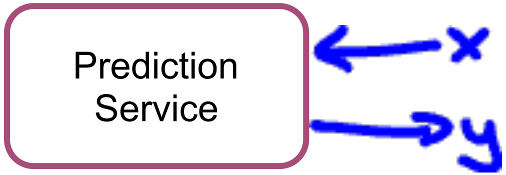
Sudden shock

How has the data changed?

Software engineering issues

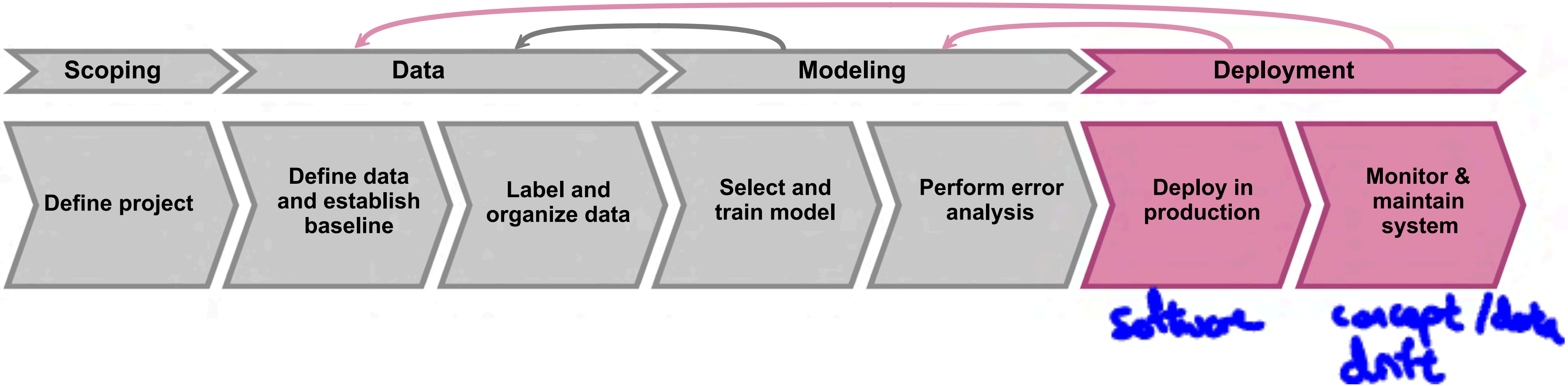
Checklist of questions

- Realtime or Batch
- Cloud vs. Edge/Browser
- Compute resources (CPU/GPU/memory)
- Latency, throughput (QPS)
- Logging
- Security and privacy


Prediction Service

500ms, 1000 QPS

First deployment vs. maintenance



Scoping

Data

Modeling

Deployment

Define project

Define data and establish baseline

Label and organize data

Select and train model

Perform error analysis

Deploy in production

Monitor & maintain system

software

concept/data drift

DeepLearning.AI

Deployment

Deployment patterns

Common deployment cases

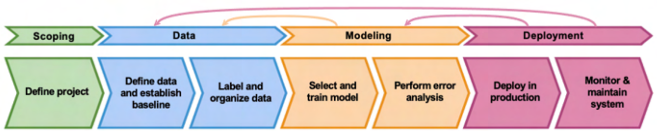
1. New product/capability
2. Automate/assist with manual task
3. Replace previous ML system

Key ideas:

- Gradual ramp up with monitoring
- Rollback

Visual inspection example

shadow mode

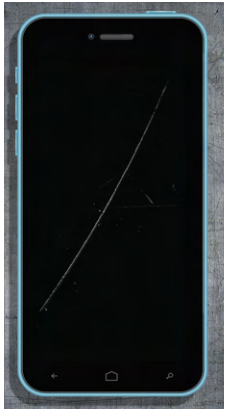


Human

✓

ML

✓

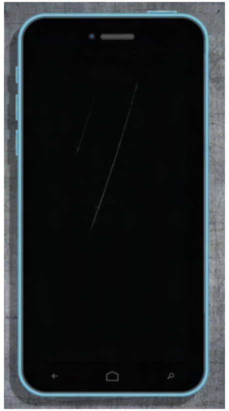


Human

✗

ML

✗



Human

✗

ML

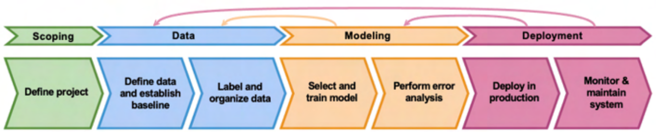
✓

ML system shadows the human and runs in parallel.

ML system's output not used for any decisions during this phase.

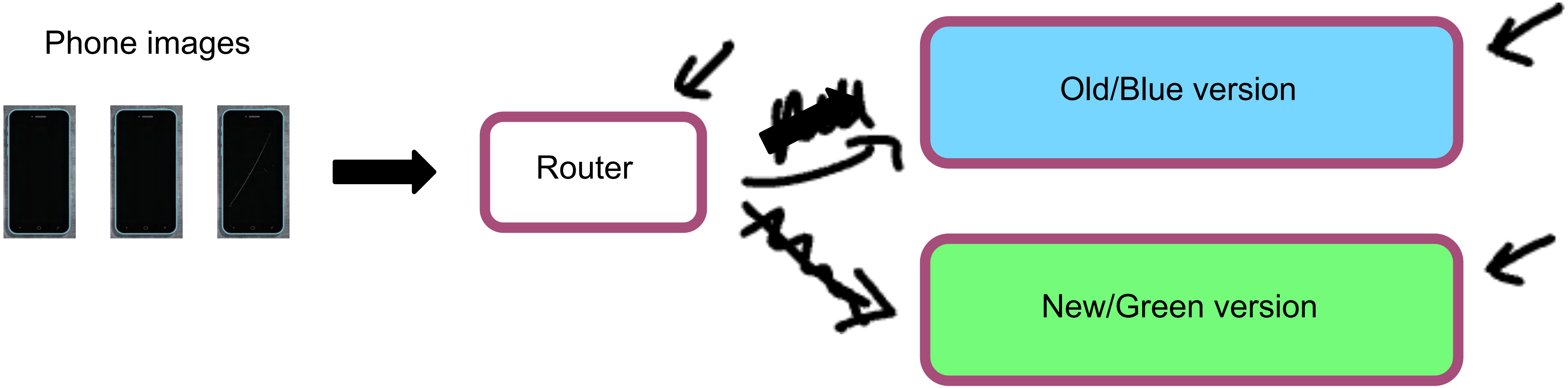
Sample outputs and verify predictions of ML system.

Canary deployment



- Roll out to small fraction (say 5%) of traffic initially.
- Monitor system and ramp up traffic gradually.

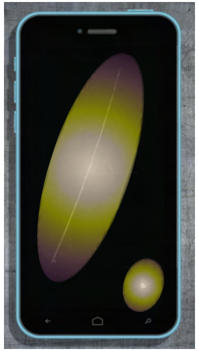
Blue green deployment

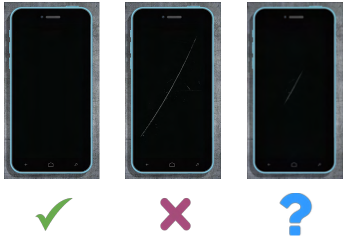


Easy way to enable rollback

Degrees of automation

human





Human

AI

Human only → Shadow mode → AI assistance → Partial automation → Full automation

human in the loop

You can choose to stop before getting to full automation.

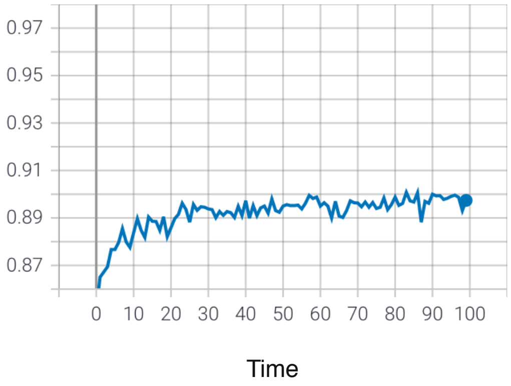
DeepLearning.AI

Deployment

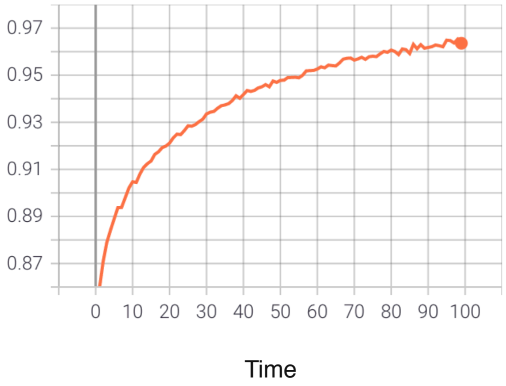
Monitoring

Monitoring dashboard

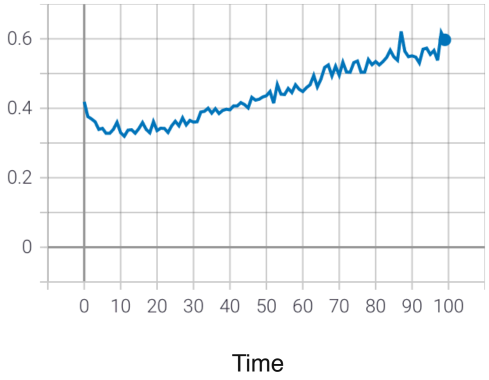
Server load



Fraction of non-null outputs



Fraction of missing input values



- Brainstorm the things that could go wrong.
- Brainstorm a few statistics/metrics that will detect the problem.
- It is ok to use many metrics initially and gradually remove the ones you find not useful.

Examples of metrics to track

Software metrics: Memory, compute, latency, throughput, server load

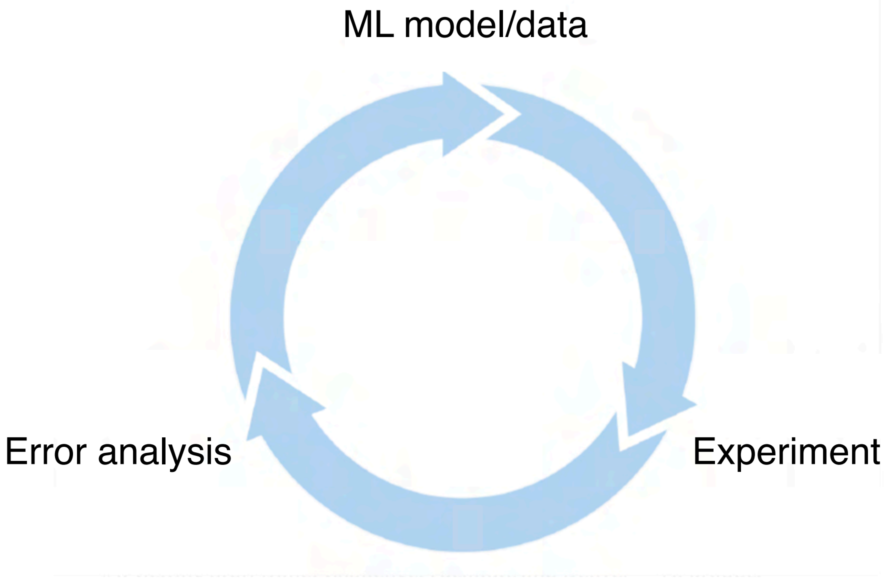
Input metrics:
x

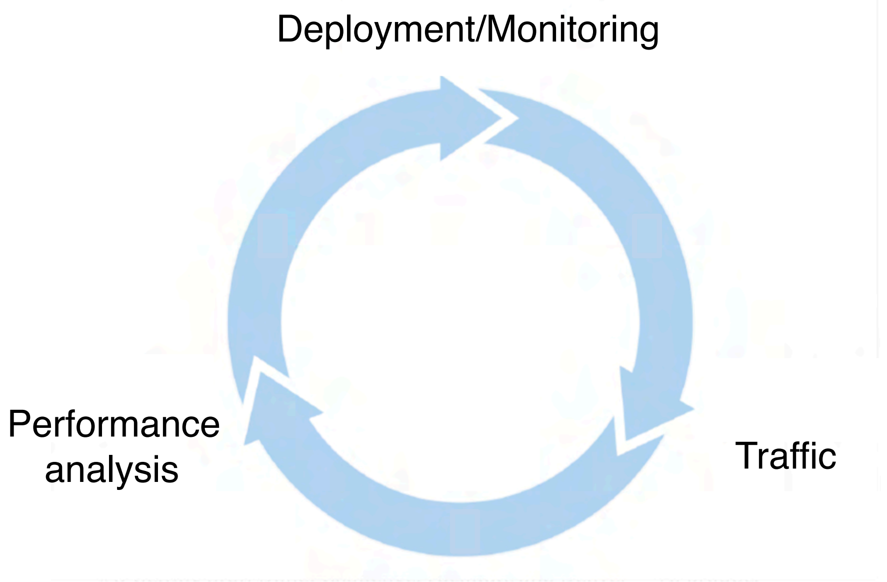
Avg input length
Avg input volume
Num missing values
Avg image brightness

Output metrics:
y

# times return " " (null)
# times user redoes search
# times user switches to typing
CTR

Just as ML modeling is iterative, so is deployment



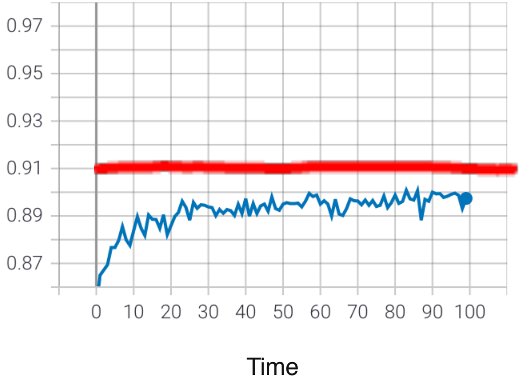


Iterative process to choose the right set of metrics to monitor.

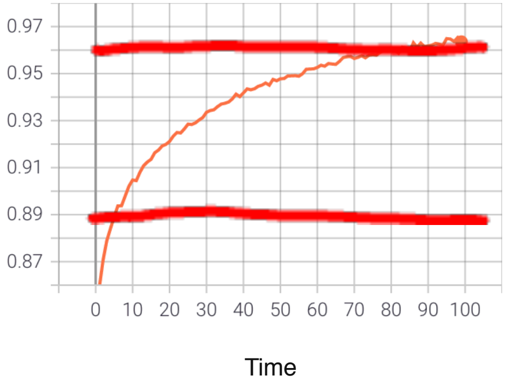
DeepLearning.AI

Monitoring dashboard

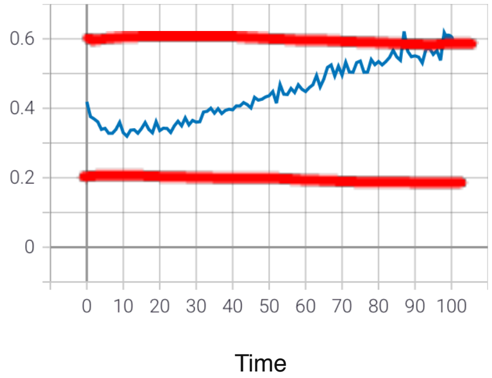
Server load



Fraction of non-null outputs

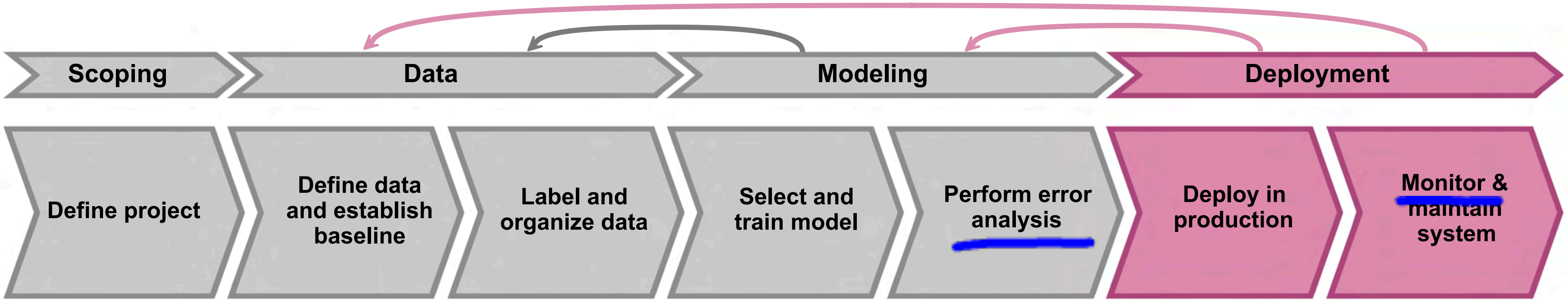


Fraction of missing input values



- Set thresholds for alarms
- Adapt metrics and thresholds over time

Model maintenance



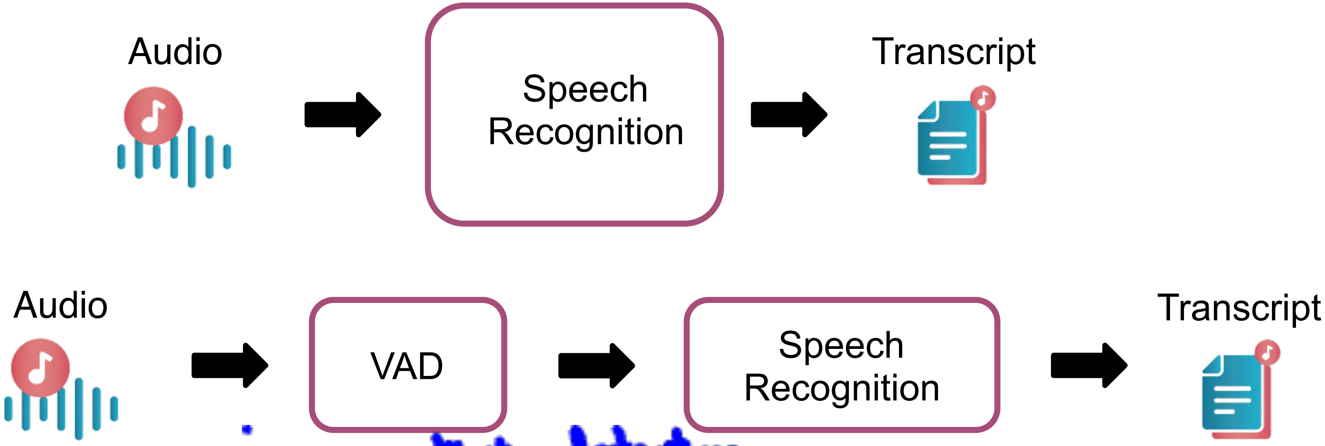
- Manual retraining
- Automatic retraining

DeepLearning.AI

Deployment

Pipeline monitoring

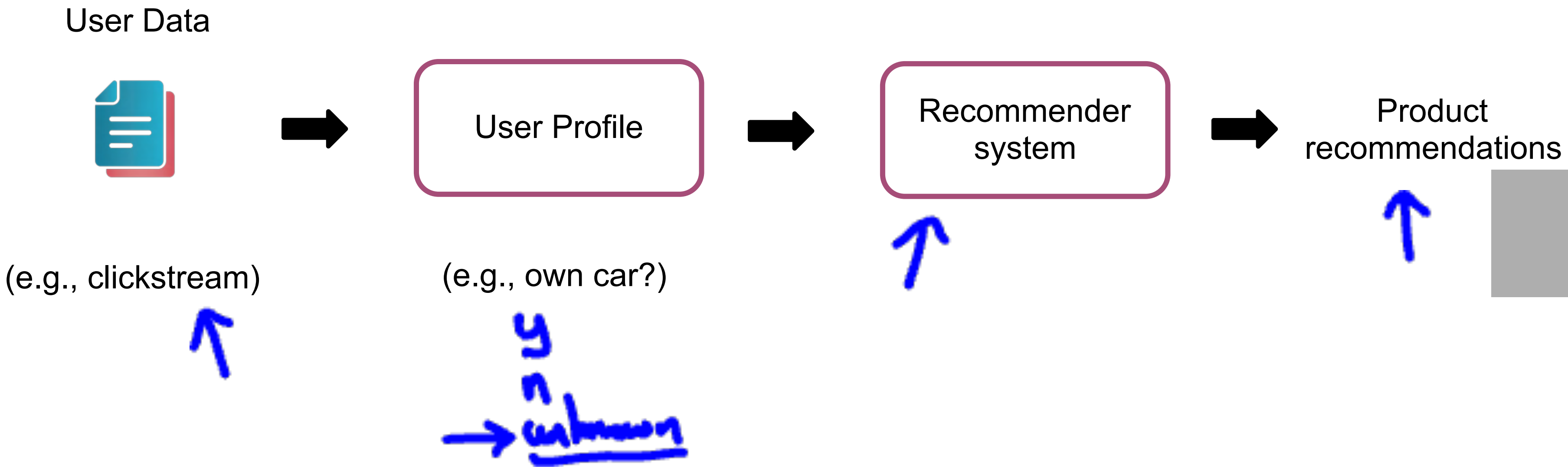
Speech recognition example



voice activity detection

Some cellphones might have VAD clip audio differently, leading to degraded performance

User profile example



User Data

User Profile

Recommender system

Product recommendations

(e.g., clickstream)

(e.g., own car?)

y

n

unknown

Metrics to monitor

Monitor

- Software metrics
- Input metrics
- Output metrics

How quickly do they change?

- User data generally has slower drift.
- Enterprise data (B2B applications) can shift fast.