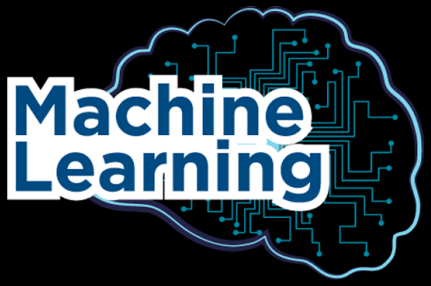
Machine Learning in Production

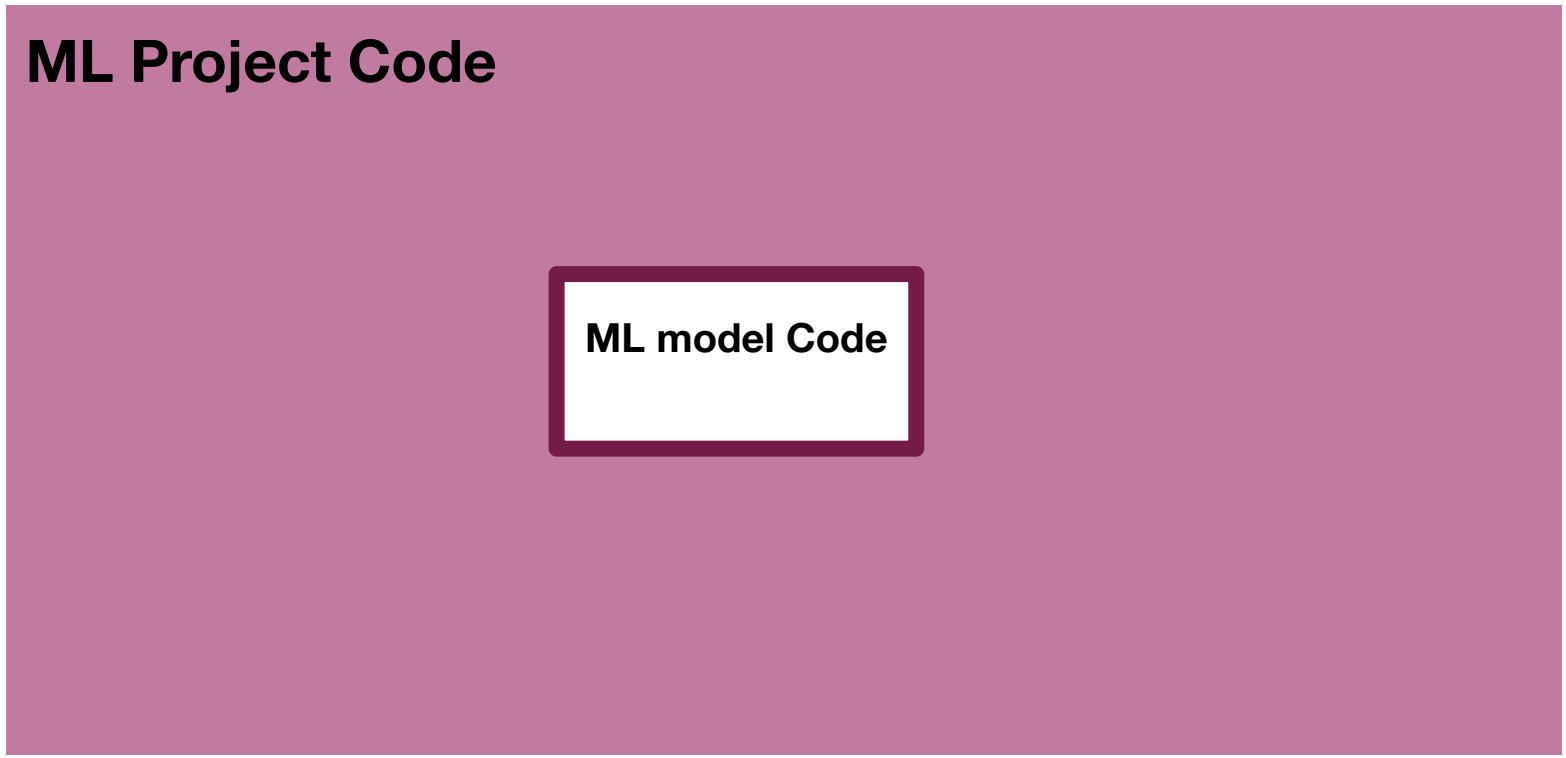


Seyed Abbas Hosseini
Sharif University of Technology

All slides are adopted from MLOPs, Andrew Ng., Deeplearning.ai and

MLSys course of Berkeley, Gonzalez

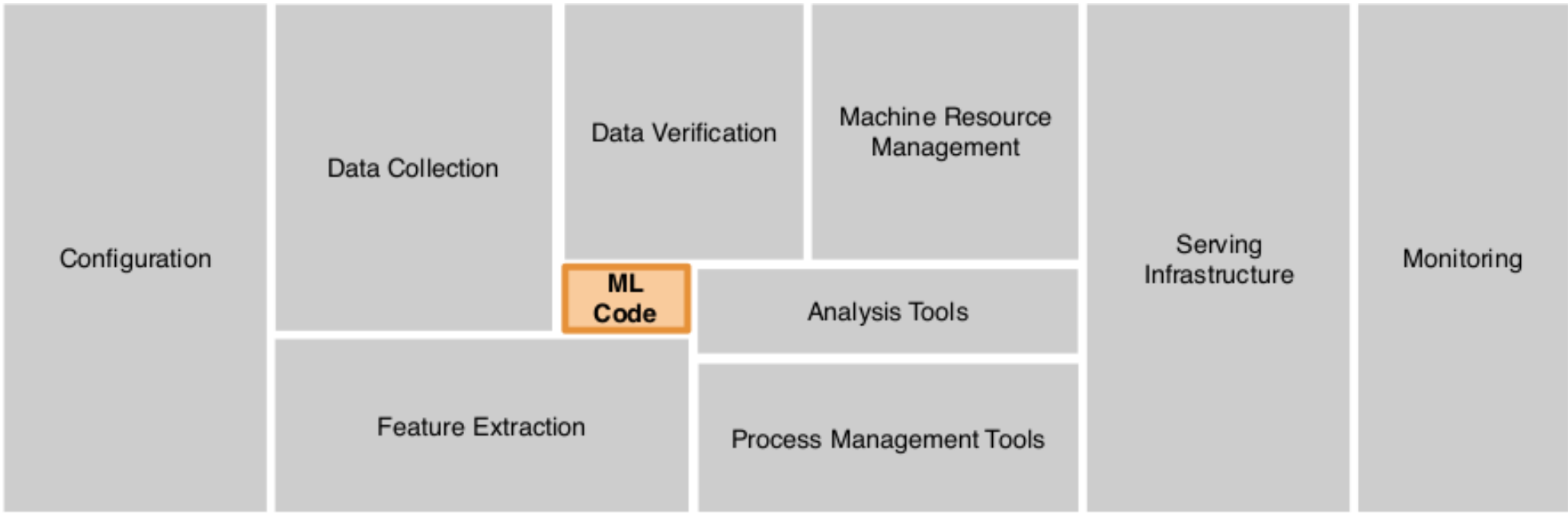
ML in production



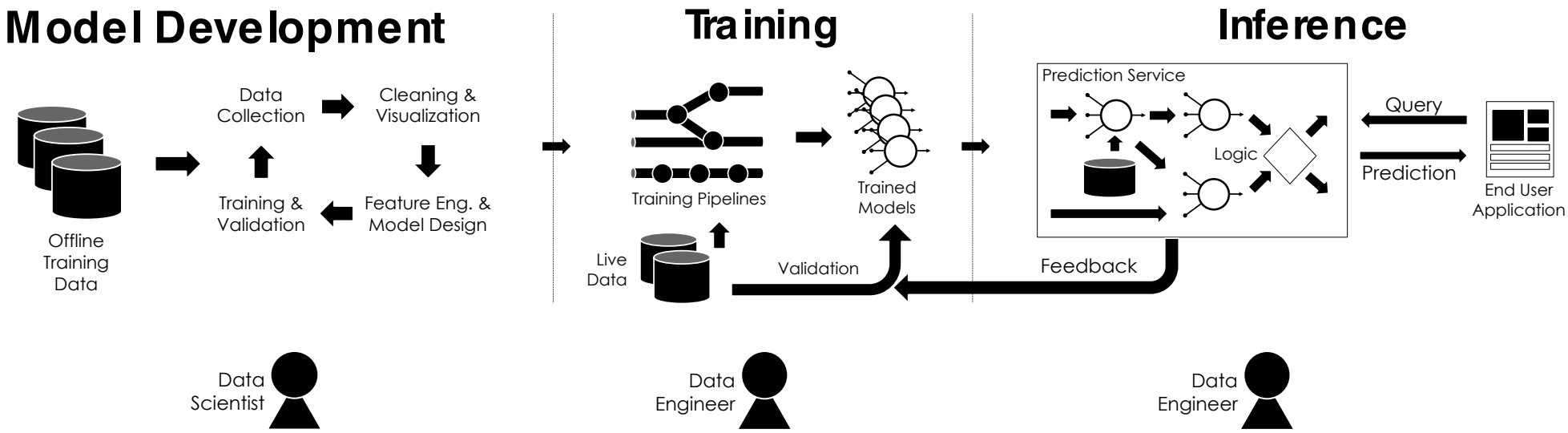
“POC to Production Gap”

ML in production

The requirements surrounding ML infrastructure



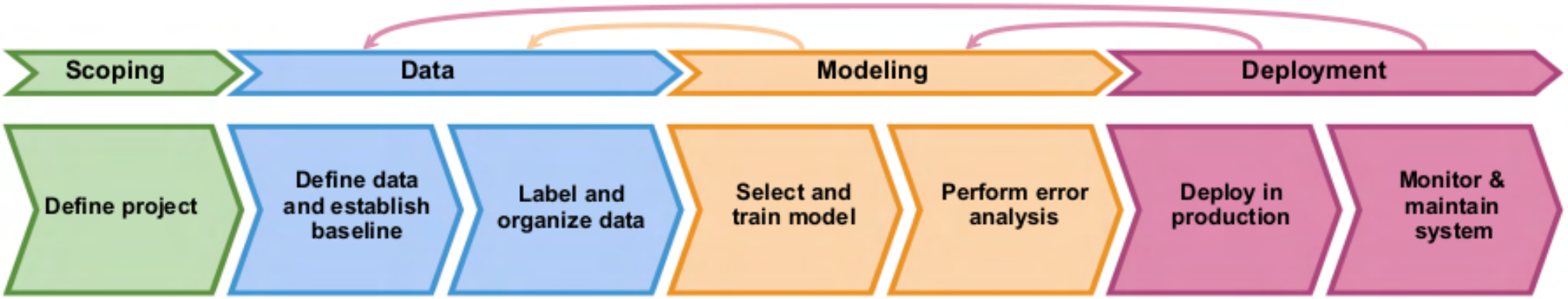
ML in Production



Machine Learning Project Lifecycle

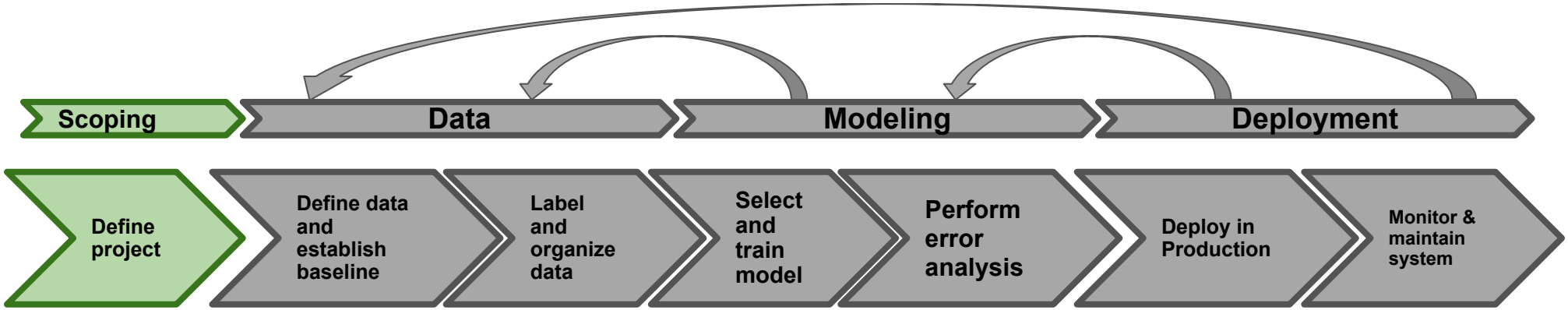
Steps of an ML project

The ML project Lifecycle



Machine Learning Project Lifecycle

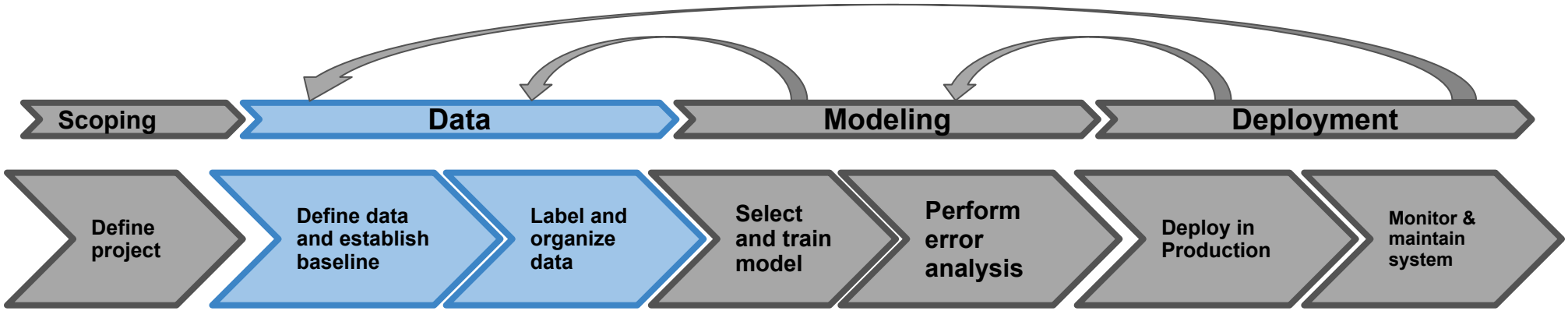
Speech recognition: Scoping stage



- Decide to work on speech recognition for voice search
- Decide on key metrics:

  Acc, Latency, Throughput
- Estimate resources and timeline

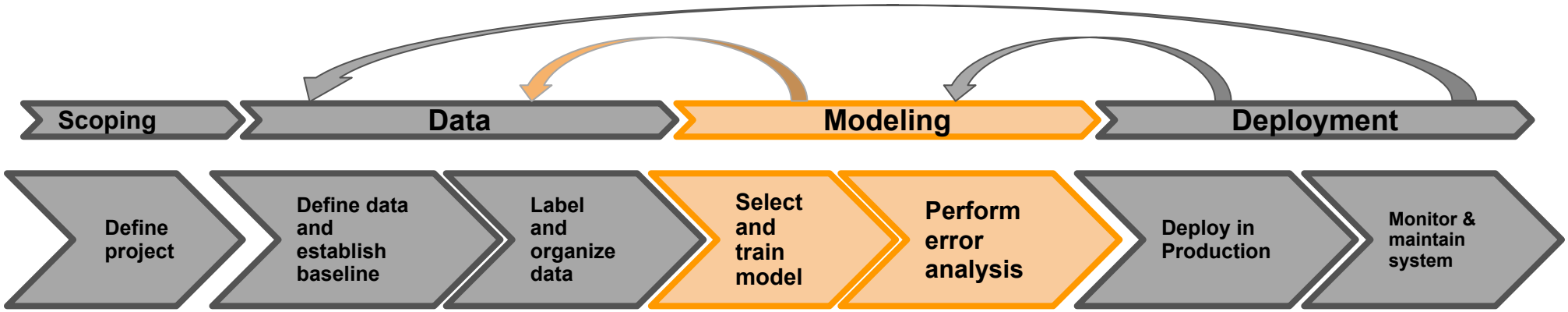
Speech recognition: Data stage



Define data

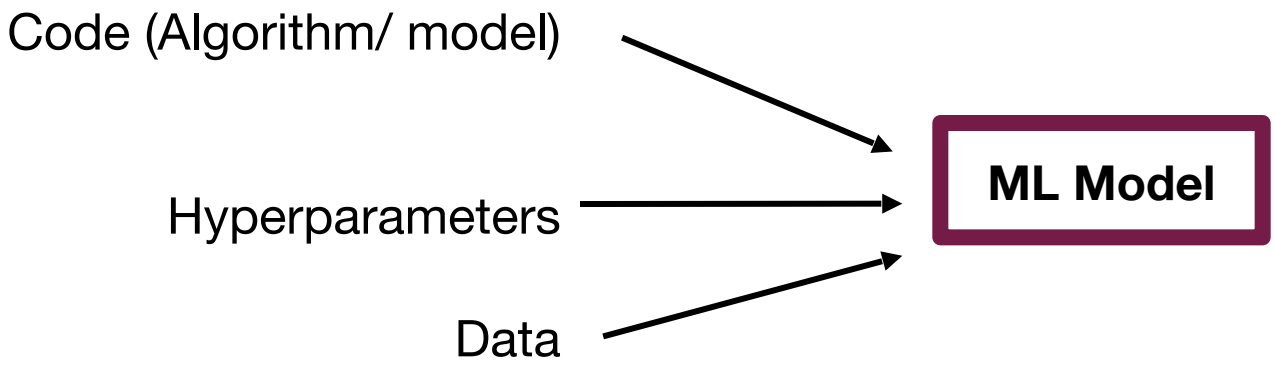
- Is the data labeled consistently?
- How much silence before/after each clip?
- How to perform volume normalization?

Speech recognition: Modeling stage

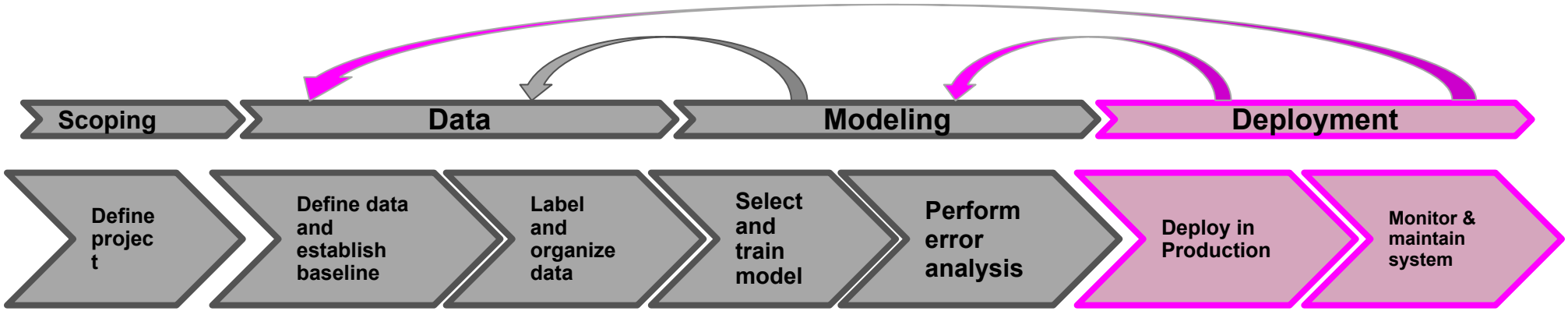


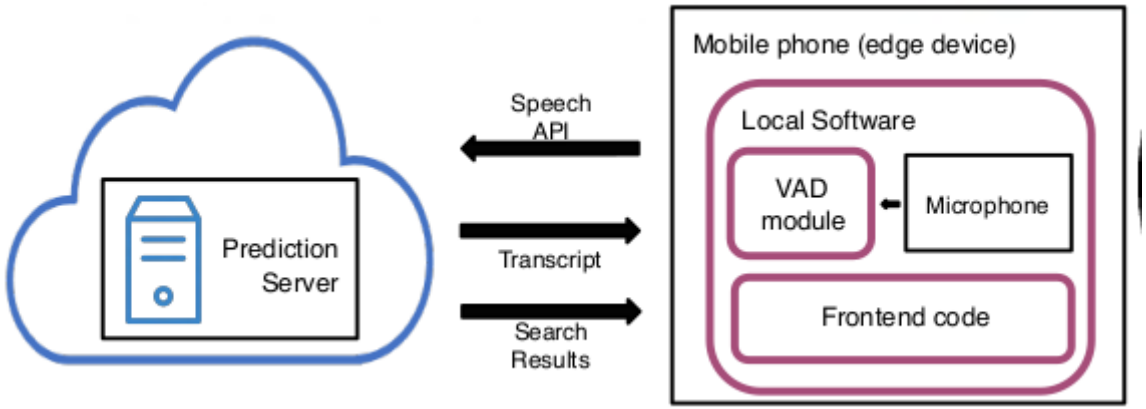
Research/ Academia

Product Team

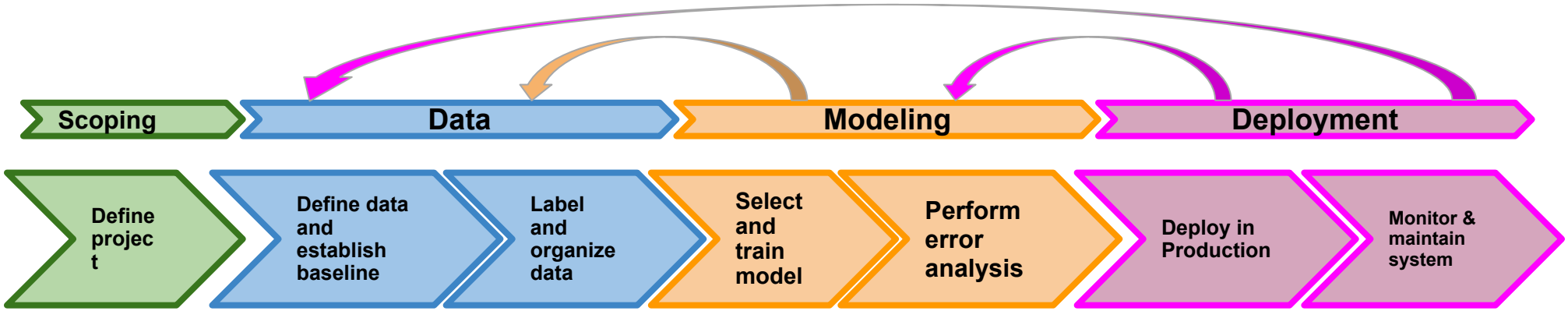


Speech recognition: Deployment stage





The Machine Learning Project Lifecycle Course outline



1. Deployment
2. Modeling
3. Data
4. 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.

Deployment

Deployment : Key challenges

Concept drift and Data drift

Speech recognition example

Training Set:

- Purchased data, historical user data with transcripts

Test set:

- Data from a few months ago

How has the data changed?

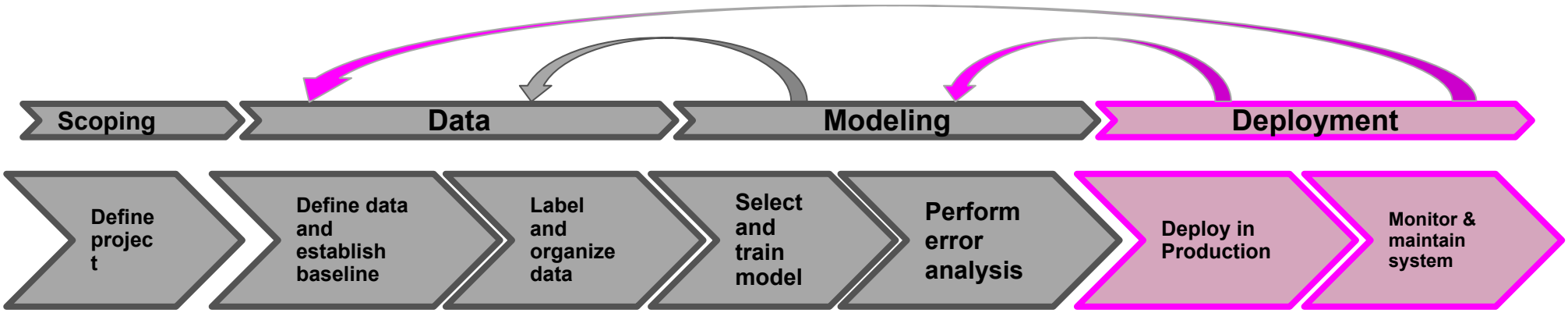
Software engineering issues

Checklist of questions

Prediction Service

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

First deployment vs. maintenance



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

ML system shadows the human and runs in parallel

ML system's output not used for any decision in this phase

Sample outputs and verify prediction of ML systems

Canary deployment

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

Blue green deployment

Phone image

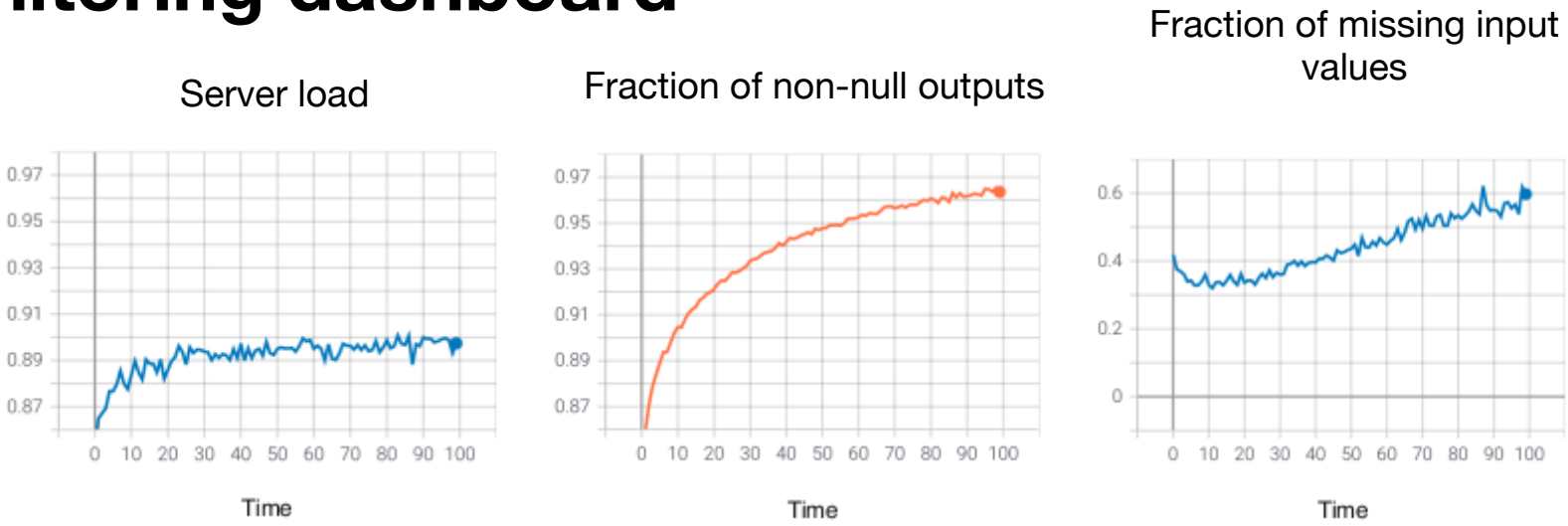
Easy way to enable rollback

Degrees of automation

You can choose to stop before getting to full automation.

Monitoring

Monitoring dashboard



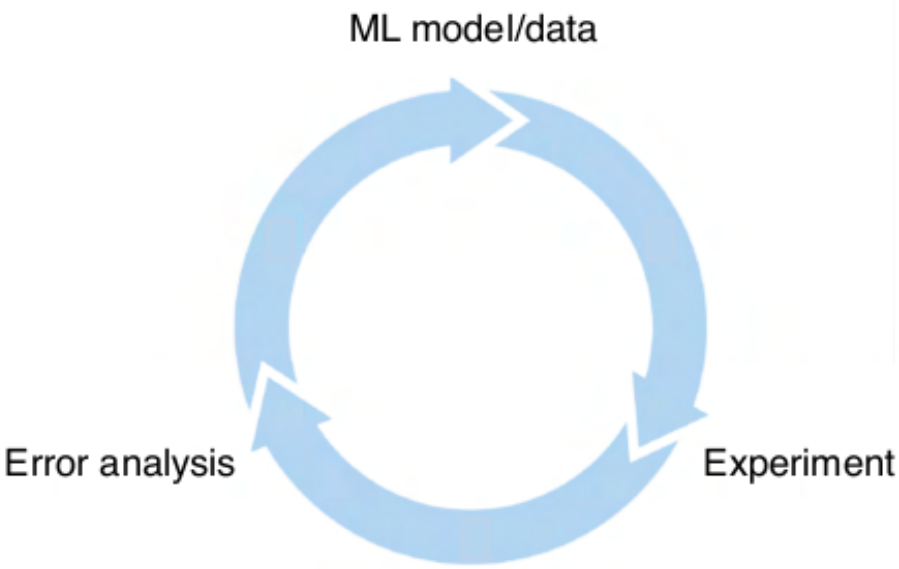
- **Brainstorm the things that could go wrong.**
- **Brainstorm a few statistics/metrics that will detect the problem.**
- **It is ok to use many metrics initially and gradually remove the ones you find not useful.**

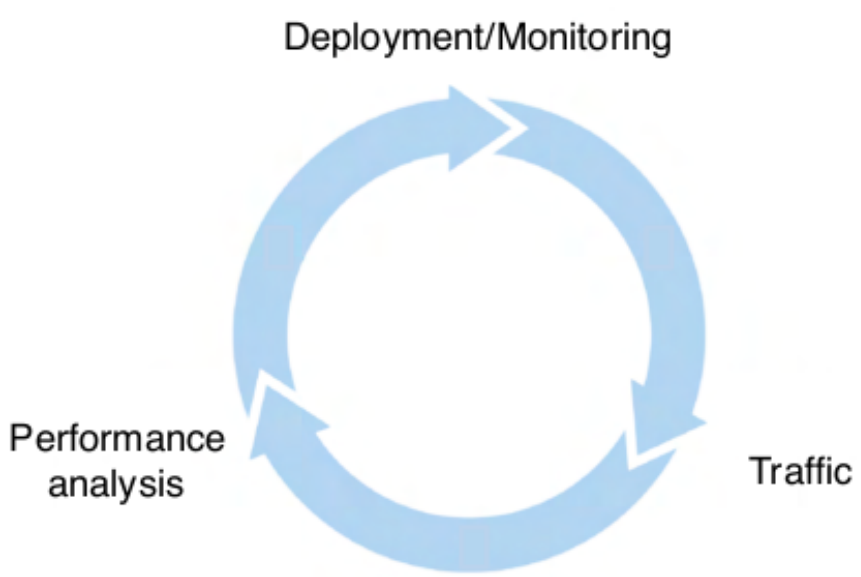
Examples of metrics to track

Software metrics	Memory, compute, latency, throughput, server load
input metrics	Avg input length Avg input volume Num missing values Avg image brightness
output metrics	# times return " " (null) # times user redoes search # times user switches to typing CTR

Iterative deployment

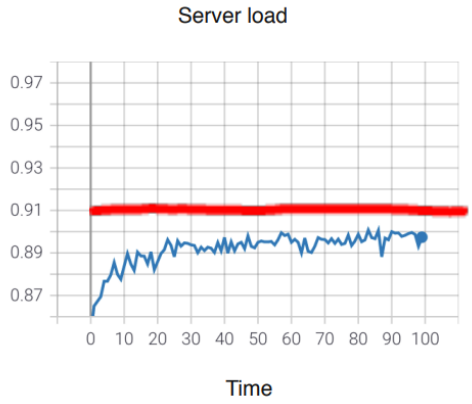
Just as ML modeling is iterative, so is deployment

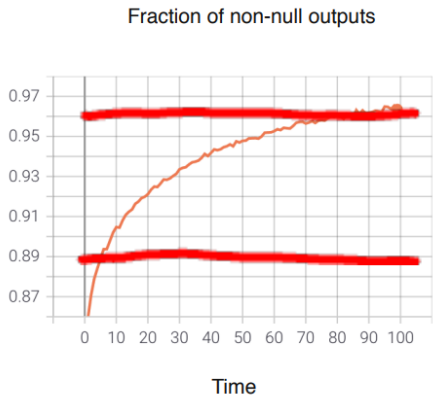


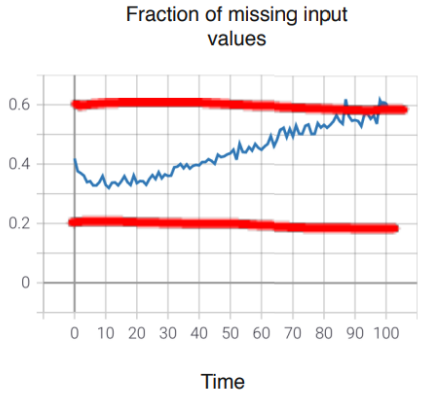


Iterative process to choose the right set of metrics to monitor.

Monitoring dashboard



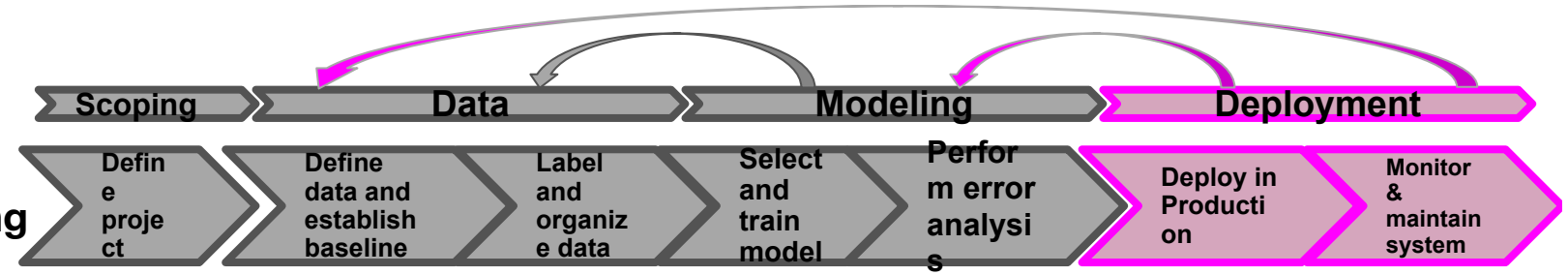




- **Set thresholds for alarms**
- **Adapt metrics and thresholds over time**

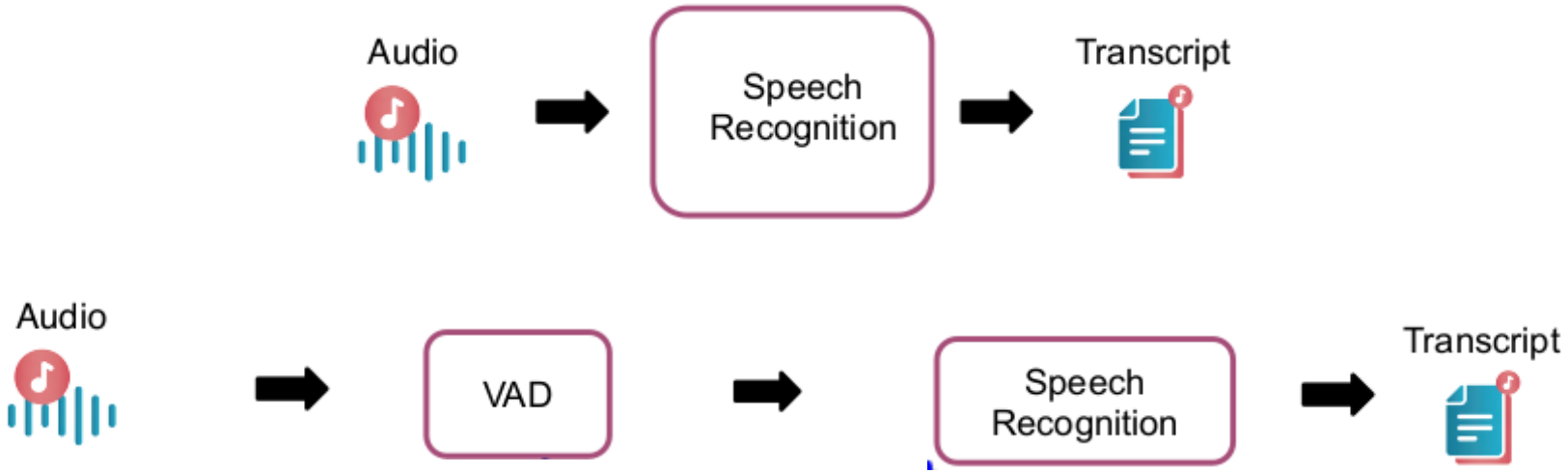
Model maintenance

- **Manual retraining**
- **Automatic retraining**



Pipeline monitoring

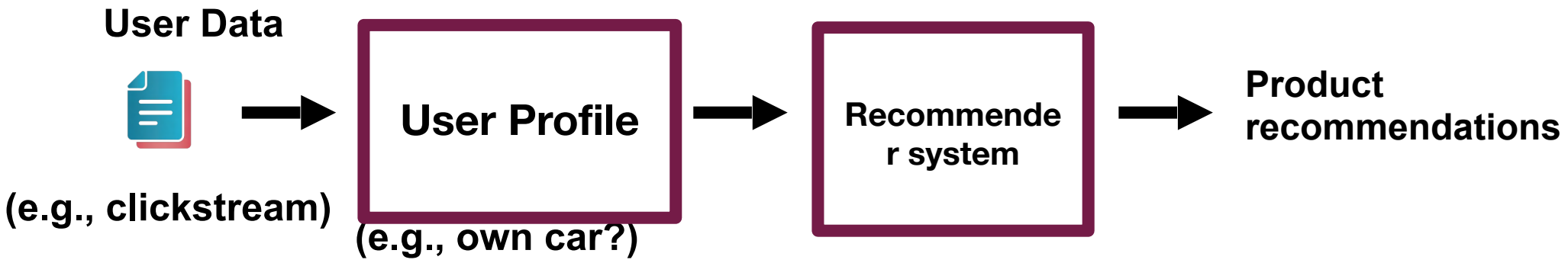
Speech recognition example



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

Pipeline monitoring

User profile example



Metrics to monitor

Monitor

- Software metrics
- input metrics
- Output metrics

How quickly do they change?

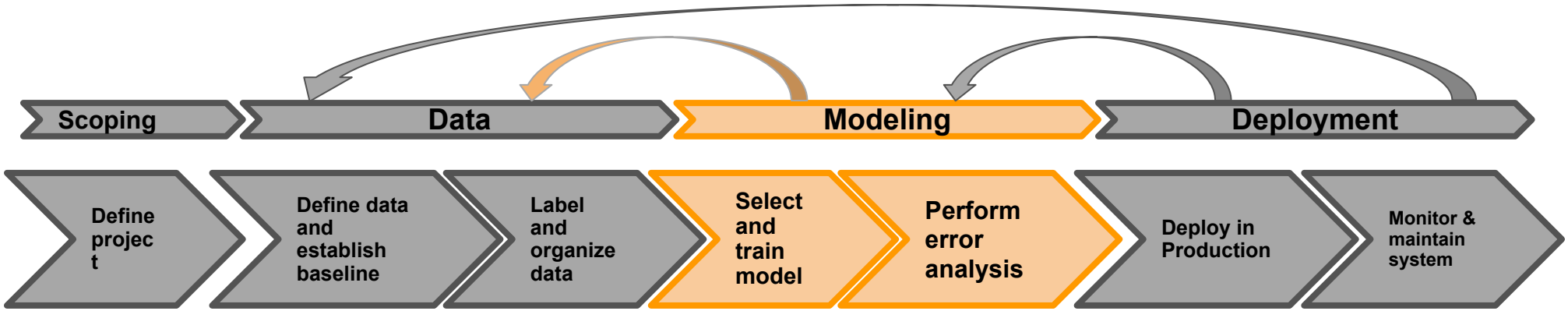
- User data generally has slower drift
- Enterprise data (B2B applications) can shift fast.

Any Questions?

Modeling

Select and train model : Modeling overview

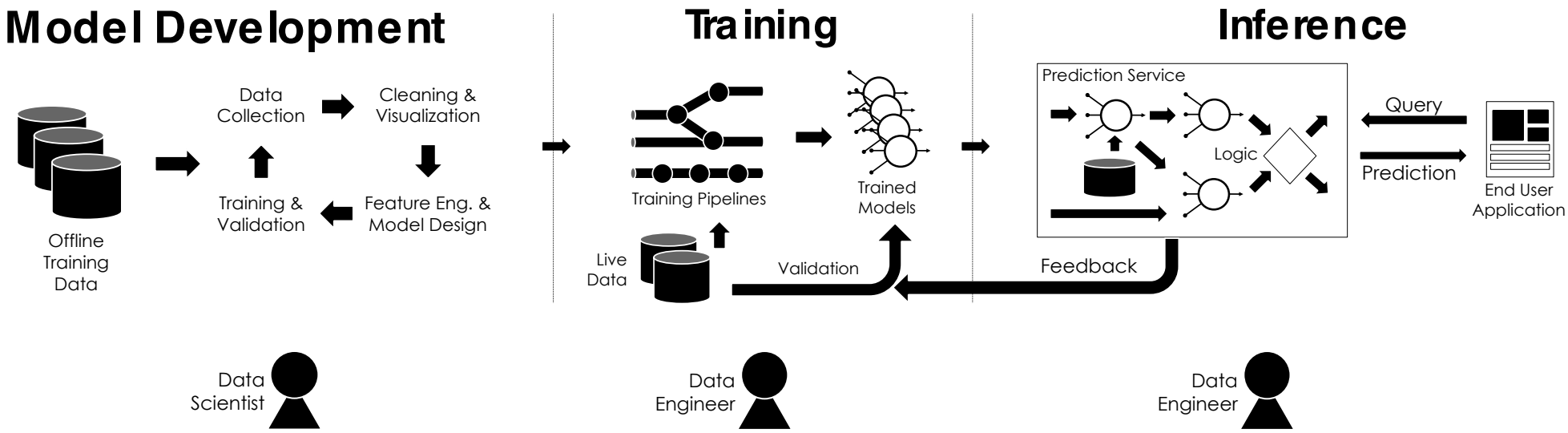
Modeling



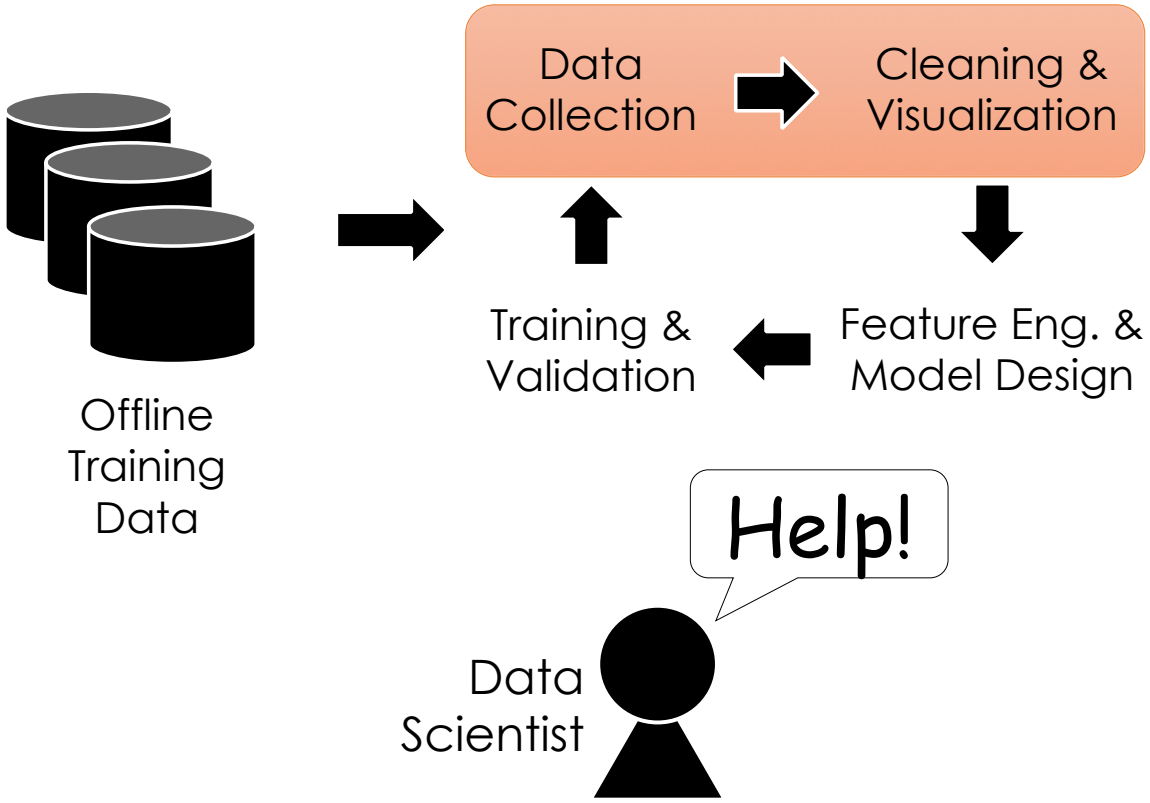
Model-centric AI development

Data-centric AI development

Model Development and Training



Model Development



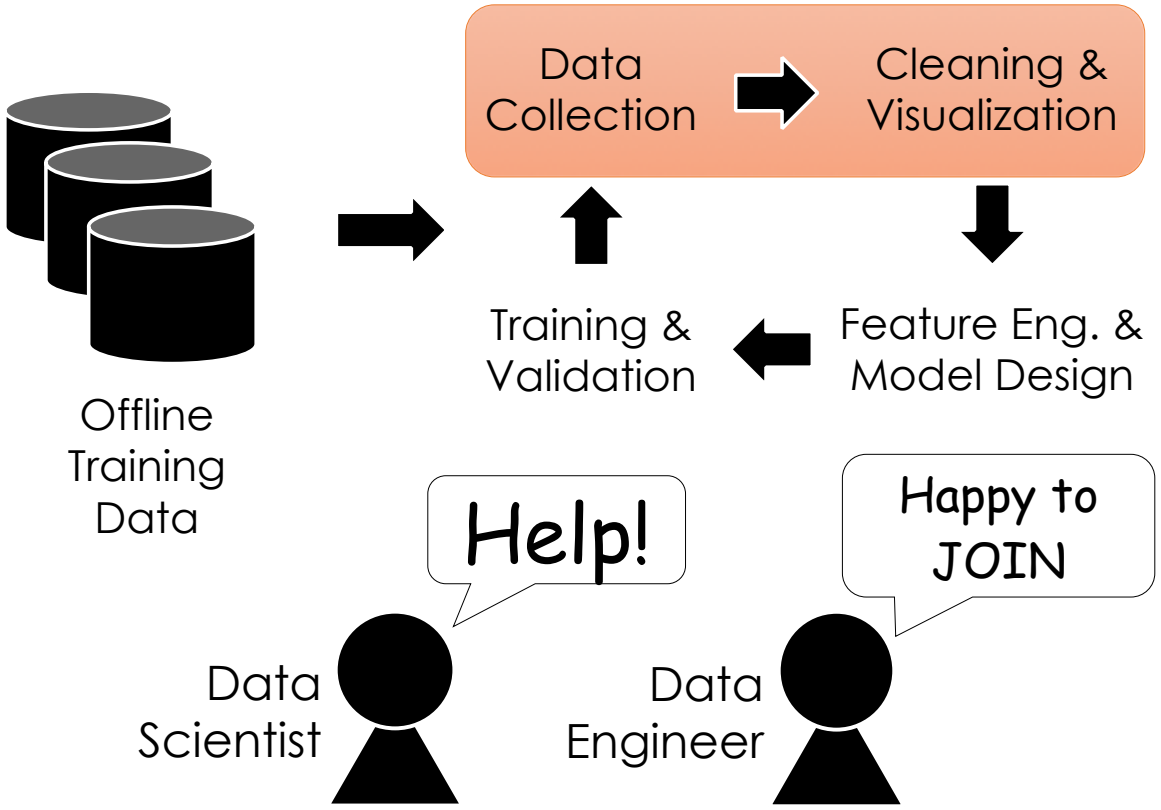
Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Model Development



Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Model Development

Big Data Borat
@BigDataBorat

Follow

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

6:47 PM - 26 Feb 2013

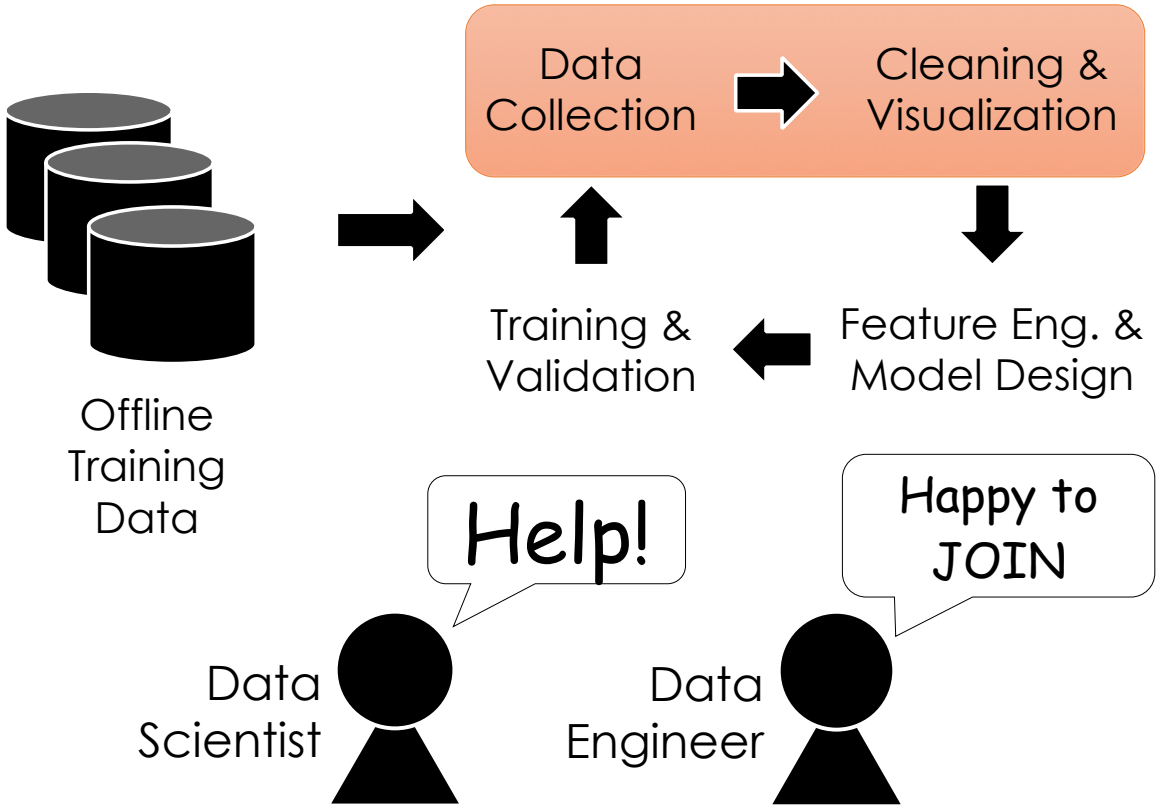
533 Retweets **330** Likes

12 533 330

Andrej Karpathy (Tesla Auto Pilot Team)

How many of you have ever worked with real data?

Model Development



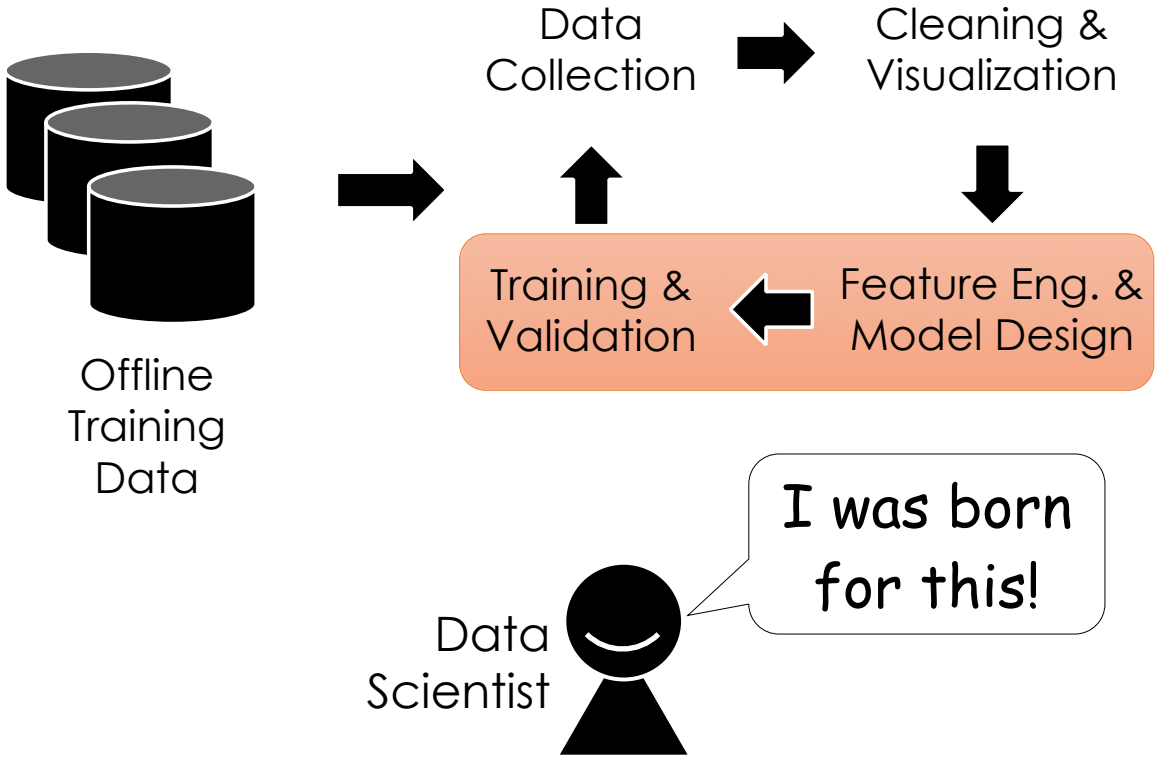
Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Model Development



Building informative **features functions**

Designing new **model architectures**

Tuning hyperparameters

Validating prediction accuracy

Features and Feature Engineering

- **Features:** properties or characteristic of the input
- **Click Prediction Example:**

Information about User and Content

Model:

$$f_\theta(x) \rightarrow y$$

Probability user will click on the content

Simple Logistic Model: $f_\theta(x) = \sigma\left(\sum_{k=1}^{d} \theta_k \phi_k(x)\right), \quad \sigma(t) = \frac{1}{1+e^{-t}}$

Features function extract numeric properties from x

Useful features?

- User Features:
  - age, gender, and click history

e.g., Recurrent NN output..

- Product Features:
  - Price, popularity, and description…

e.g., Language Model Embedding

- Combined (Cross) features:
  - I(20 < age < 30, male, "xbox" in desc)…

Hand coded features

Additional Notes on Features

- **Feature Joins:** combine multiple data source in a feature
- **Feature Reuse**: good features can aid in many tasks
  - *Example:* product embeddings, user tags, …
- **Predictions as Features:** predictions for one task (e.g., products in an image) can be useful features for another (e.g., ad targeting)
- **Feature Tables/Caches:** features are often pre-computed and cached
  - Requires tracking data and compute and feature versions
- **Dynamic Features:** features can often be modified faster than models
  - Useful for addressing fast changing dynamics (e.g., user preferences can be encoded in click history features).
  - **Issue:** resulting potential covariate shift can be problematic

Hyperparameters

- the **parameters** and more generally **configuration details** that are not directly determined through training
  - set by hand or tuned using cross validation
  - why not learn directly?
- Find the Hyperparameters:

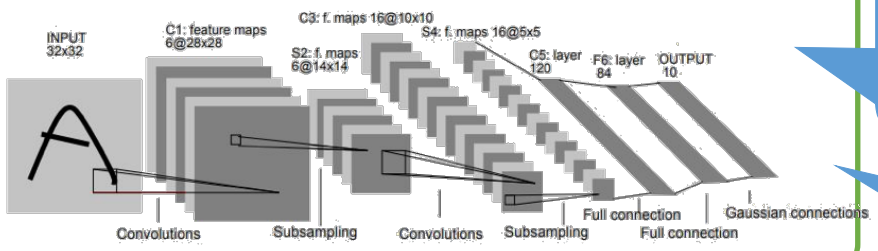
Objective:

$$\arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L_{\alpha}\left(f_{\theta}(x_i), y_i\right) + \lambda R(\theta)$$

Training Algorithm

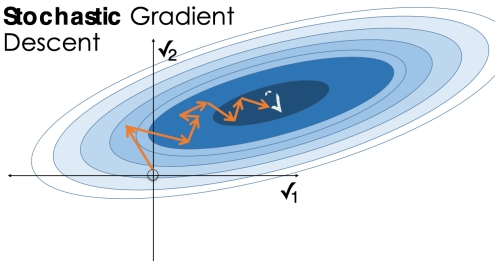
$$u^{(t)} \leftarrow \beta u^{(t-1)} + \eta \sum_{i \in \mathcal{B}} \nabla_{\theta} \left(L_{\alpha}(f_{\theta}(x_i), y_i)\right)\Big|_{\theta^{(t)}}$$

Architecture:



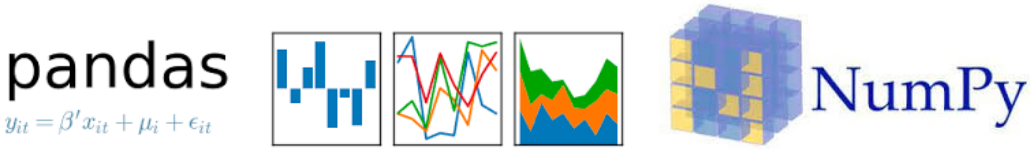
Architecture is sometimes treated as separate from hyperparameters

Can be learned…



Model Development Technologies

Model Development



What is the output of model development?

Offline Training Data

Data Collection

Cleaning & Visualization

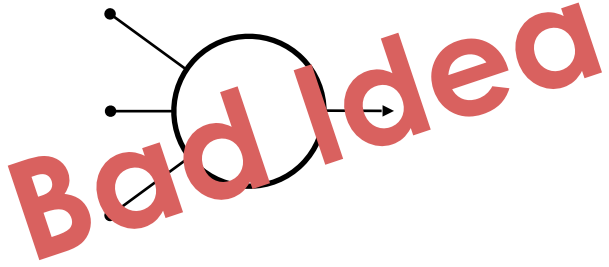
Feature Eng. & Model Design

Training & Validation

Reports & Dashboar

(insights …)

Trained Model



Why is it a **Bad Idea** to directly produce **trained models** from **model development**?

With just a trained model we are **unable to**

1. **retrain** models with new data
2. track data and code for **debugging**
3. capture **dependencies** for deployment
4. audit training for **compliance** (e.g., GDPR)

What is the output of model development?

Offline Training Data

Data Collection

Cleaning & Visualization

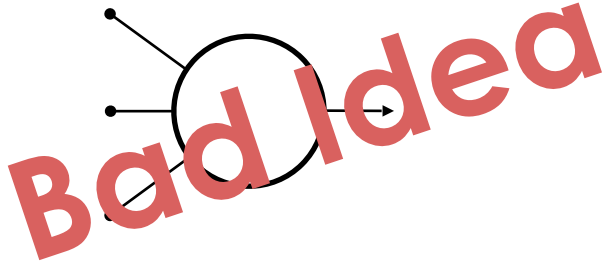
Feature Eng. & Model Design

Training & Validation

Reports & Dashboar

(insights …)

Trained Model



What is the output of model development?

Offline Training Data

Data Collection

Cleaning & Visualization

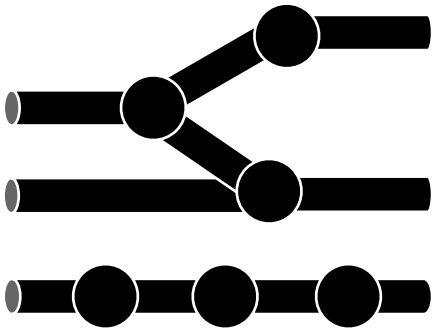
Feature Eng. & Model Design

Training & Validation

Reports & Dashboar

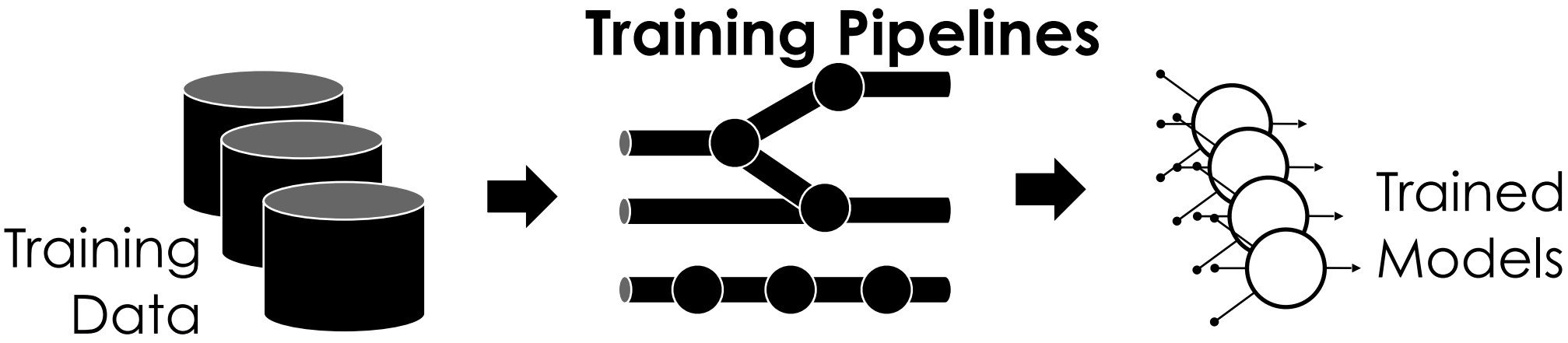
(insights …)

Training Pipelines



Training Pipelines Capture the **Code** and **Data Dependencies**

➢ Description of how to train the model from data sources



Software Engineering Analogy	
	Training Pipelines → Code
	Trained Models → Binaries

What is the output of model development?

Offline Training Data

Data Collection

Cleaning & Visualization

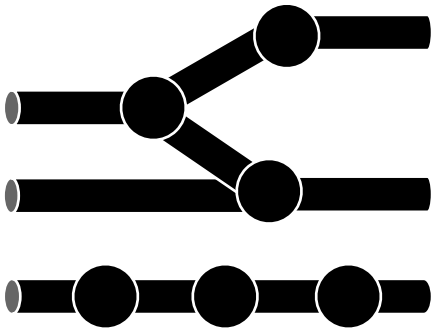
Feature Eng. & Model Design

Training & Validation

Reports & Dashboar

(insights …)

Training Pipelines



Experiment tracking

Experiment tracking

What to track?

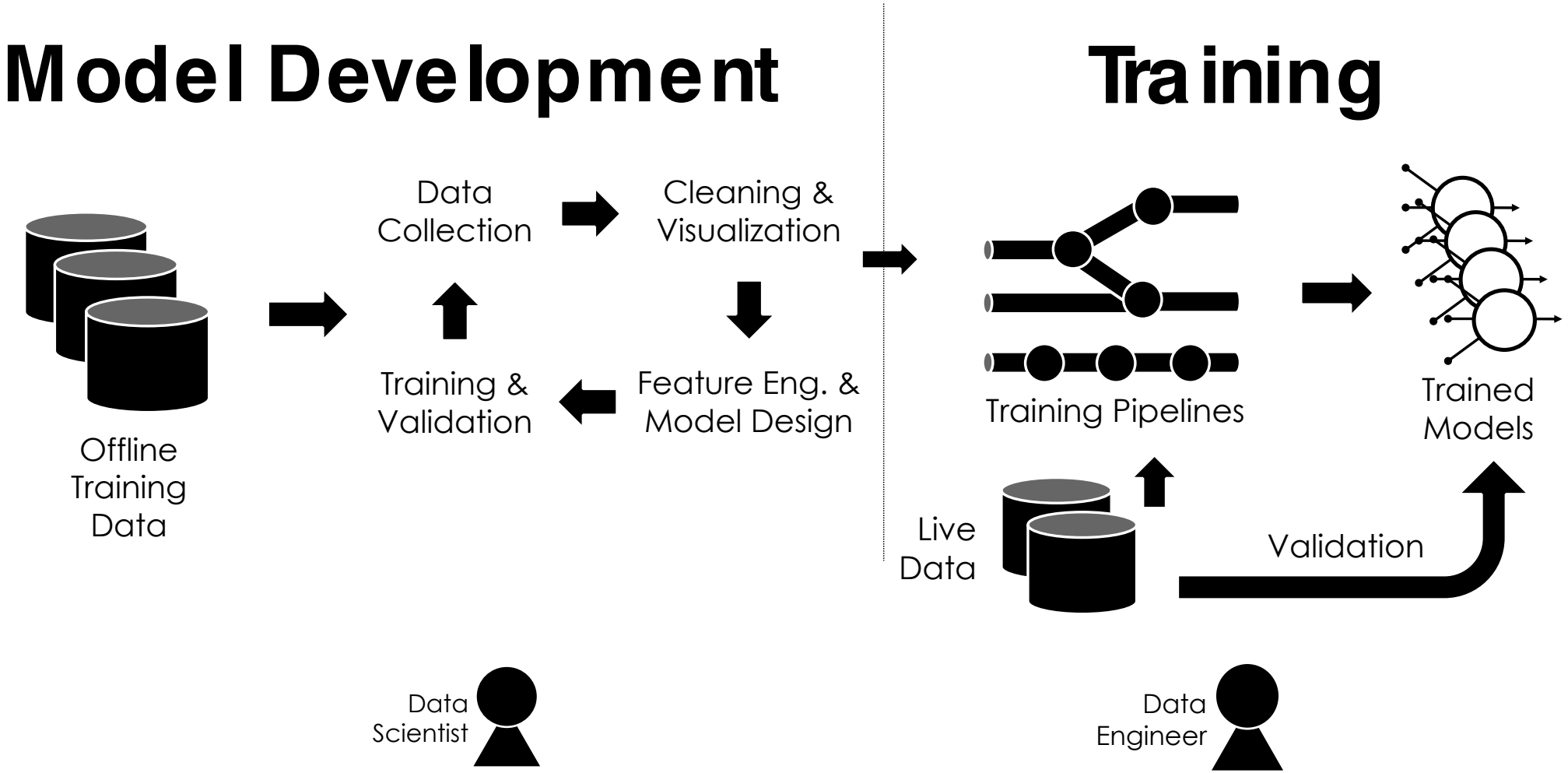
- Algorithm/code versioning
- Dataset used
- Hyperparameters
- Result

Tracking tools

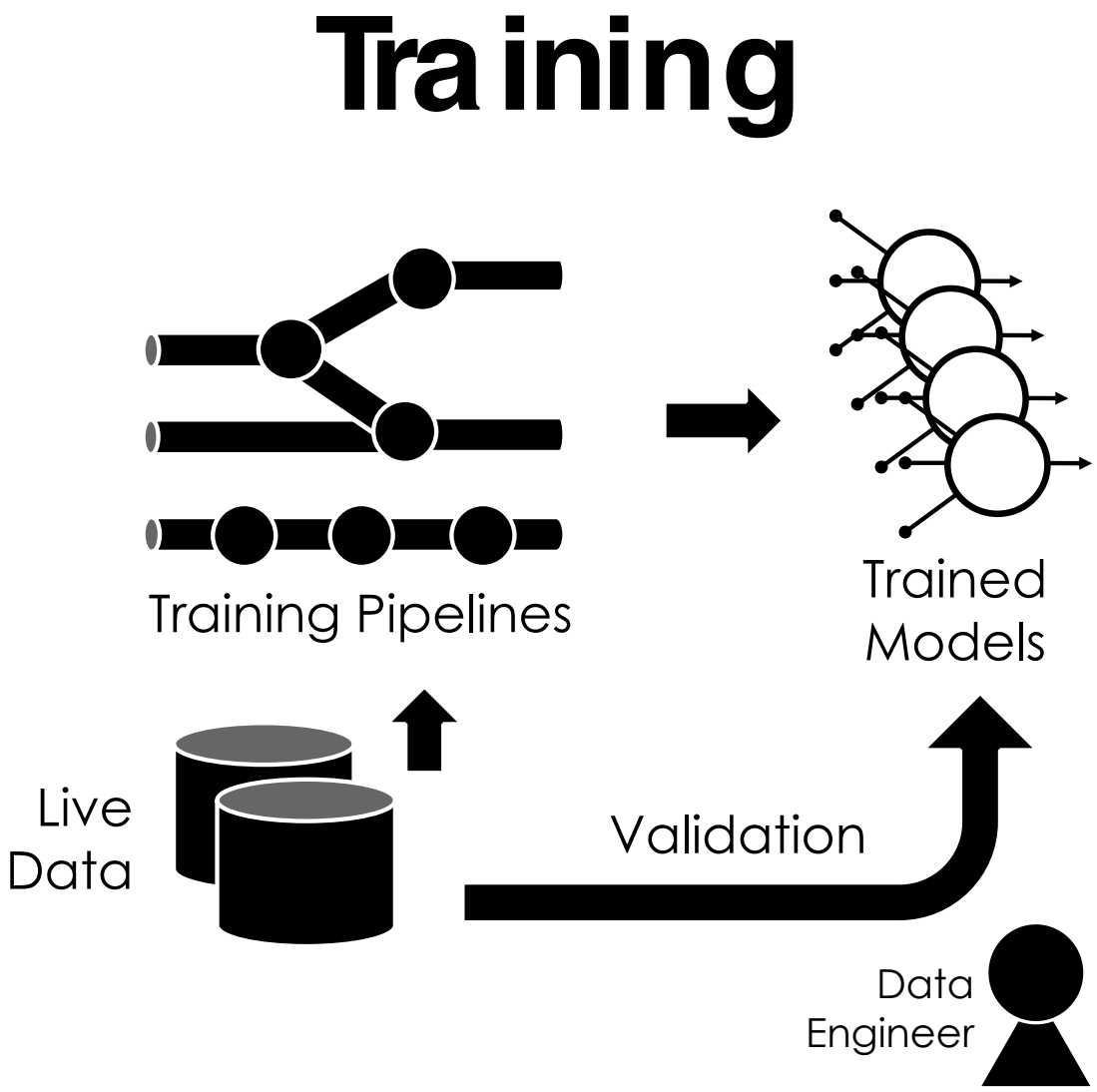
- Text files Spreadsheet
- Experiment tracking system

Desirable features

- Data needed to replicate results
- In-depth analysis of experiment results
- Perhaps also: Resource monitoring, visualization, model error analysis


Model Development
Training
Data Collection
Cleaning & Visualization
Feature Eng. & Model Design
Training & Validation
Offline Training Data
Training Pipelines
Trained Models
Live Data
Validation
Data Scientist
Data Engineer

Training



Training models **at scale** on **live data**

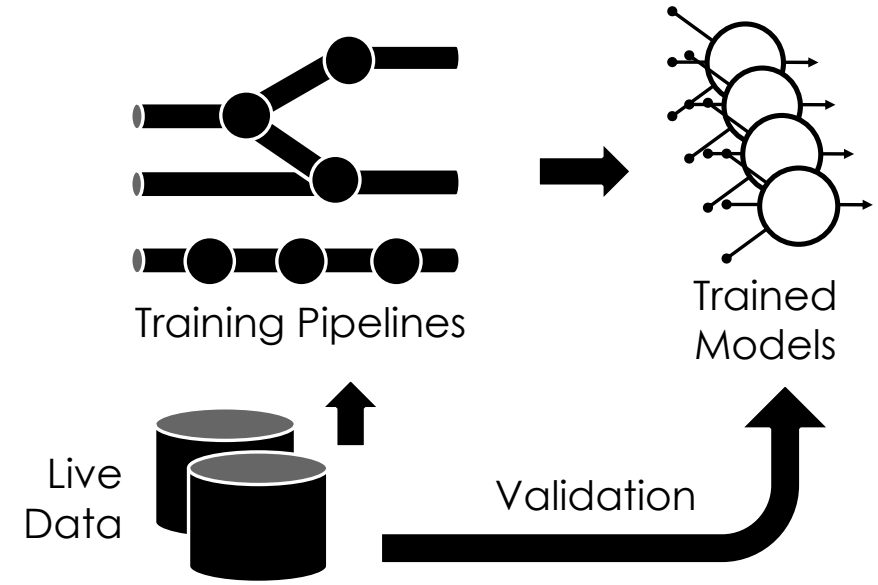
Retraining on new data

Automatically **validate** prediction accuracy

Manage model **versioning**

Requires **minimal expertise** in machine learning

Training Technologies



Workflow Management:

Apache Airflow

Luigi

Azkaban
Open-source Workflow Manager

OOZIE

Scalable Training:

PYTORCH

mxnet

TensorFlow

HOROVOD

APACHE Spark™

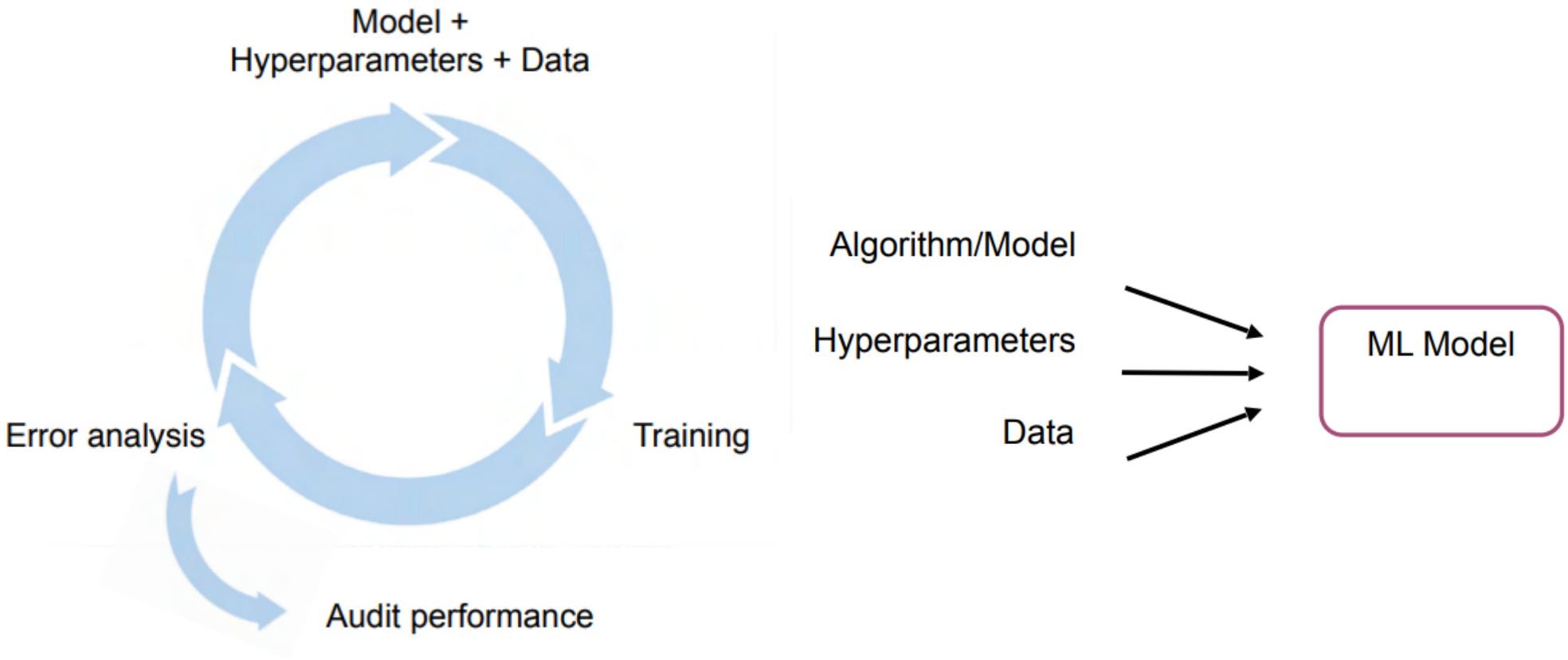
dmlc XGBoost

Model Development

- **AI system = Code + Data**

  (algorithm/model)

Model development is an iterative process



Model Development Challenges: Unfortunate conversation in many companies

MLE: "I did well on the test set!"

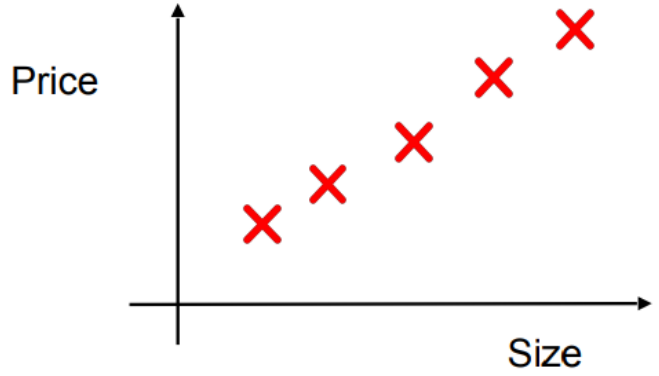
Product Owner: "But this doesn't work for my application"

MLE: "But... I did well on the test set!"

Model Development Challenges

1. Doing well on training set (usually measured by average training error).

2. Doing well on dev/test sets.



3. Doing well on business metrics/project goals.

Select and train model : Why low average test error isn't good enough

Performance on disproportionately important examples

Web Search example

"Apple pie recipe" "Latest movies"

"Wireless data plan" "Diwali festival"

} Informational and Transactional queries

"Stanford" "Reddit" "Youtube"

} Navigational queries

Model Development Challenges

Performance on key slices of the dataset

Example: ML for loan approval

Make sure not to discriminate by ethnicity, gender, location, language or other protected attributes.

Example: Product recommendations from retailers

Be careful to treat fairly all major user, retailer, and product categories.

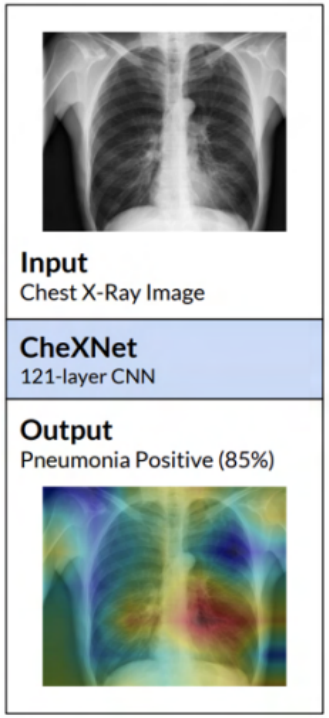
Model Development Challenges

Skewed data distribution

print("0")

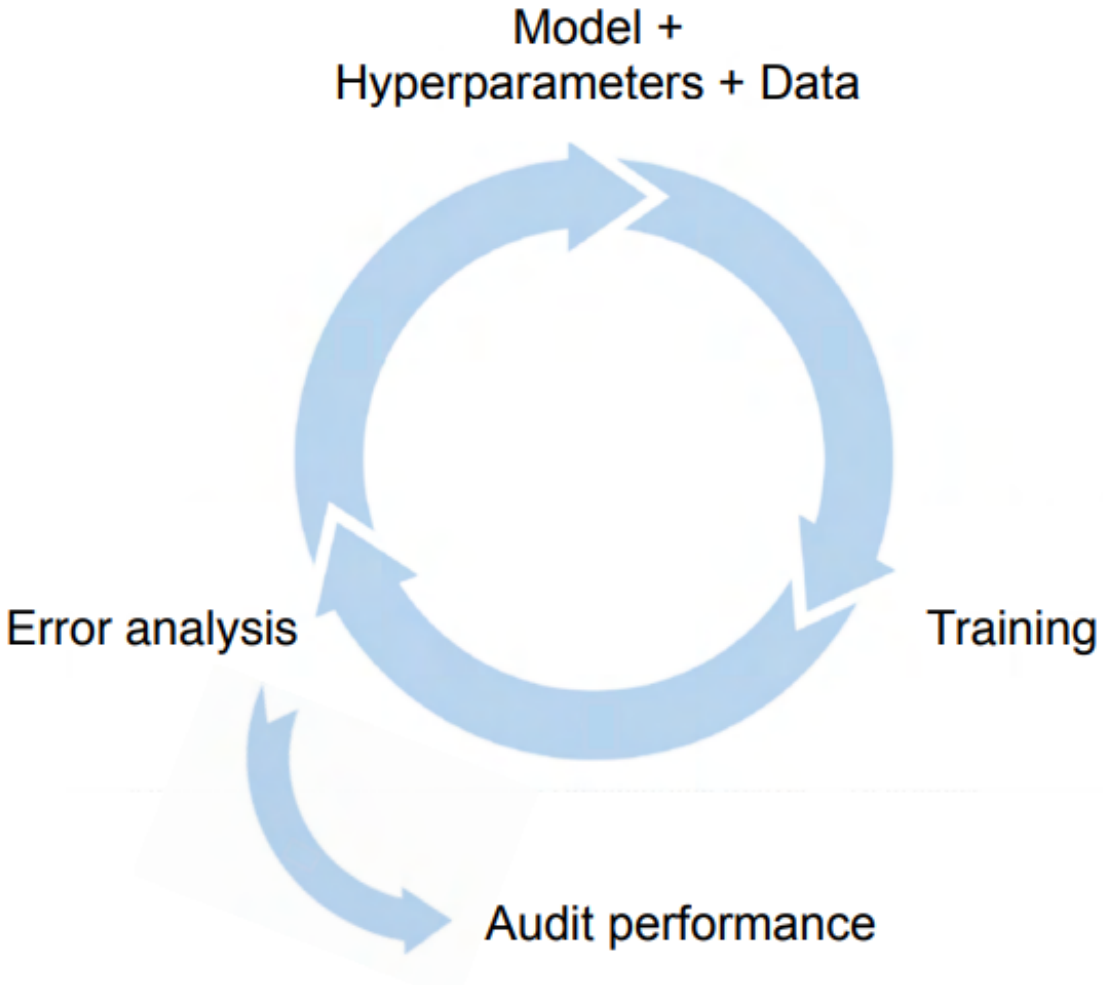
Accuracy in rare classes

Condition	Performance
Effusion	0.901
Edema	0.924
Mass	0.909
Hernia	0.851



Select and train model : Tips for getting started

ML is an iterative process



Getting started on modeling

- Literature search to see what's possible.
- Find open-source implementations if available.
- A reasonable algorithm with good data will often outperform a great algorithm with not so good data.

Sanity-check for code and algorithm

- Try to overfit a small training dataset before training on a large one.

- " Example #1: Speech recognition

- " Example #2: Image segmentation

- " Example #3: Image classification

Error Analysis

Example

Propose tags

Visual inspection:

- Specific class labels (scratch, dent, etc.)
- Image properties (blurry, dark background, light background, reflection….)
- Other meta-data: phone model, factory

Product recommendations:

- User demographics
- Product features

Error Analysis: Propose Tags

Speech recognition example

Different types of speech input:

- **Car noise**
- **Plane noise**
- **Train noise**
- **Machine noise**
- **Cafe noise**
- **Library noise**
- **Food court noise**

Error analysis and performance auditing : Error analysis example

Speech recognition example

Example	Label	Prediction	Car Noise	People Noise	Low Bandwidth
1	"Stir fried lettuce recipe"	"Stir fry lettuce recipe"	✓		
2	"Sweetened coffee"	"Swedish coffee"		✓	✓
3	"Sail away song"	"Sell away some"		✓	
4	"Let's catch up"	"Let's ketchup"	✓	✓	✓

Useful metrics for each tag

- What fraction of errors has that tag?
- Of all data with that tag, what fraction is misclassified?
- What fraction of all the data has that tag?
- How much room of improvement is there in that tag?

Ways to establish a baseline

- **" Human level performance (HLP)**
- **Literature search for state-of-the-art/open source**
- **" Older system "**

Baseline gives an estimate of the irreducible error / Bayes error and indicates what might be possible.

Select and train model : Establish a baseline

Establishing a baseline level of performance

 **Speech recognition example:**

Type	Accuracy	Human level performance
Clear Speech	94%	95%
Car Noise	89%	93%
People Noise	87%	89%
Low Bandwidth	70%	70%

Prioritizing what to work on

Decide on most important categories to work on based on:

- How much room for improvement there is.
- How frequently that category appears.
- How easy is to improve accuracy in that category.
- How important it is to improve in that category.

Error analysis and performance auditing : Prioritizing what to work on

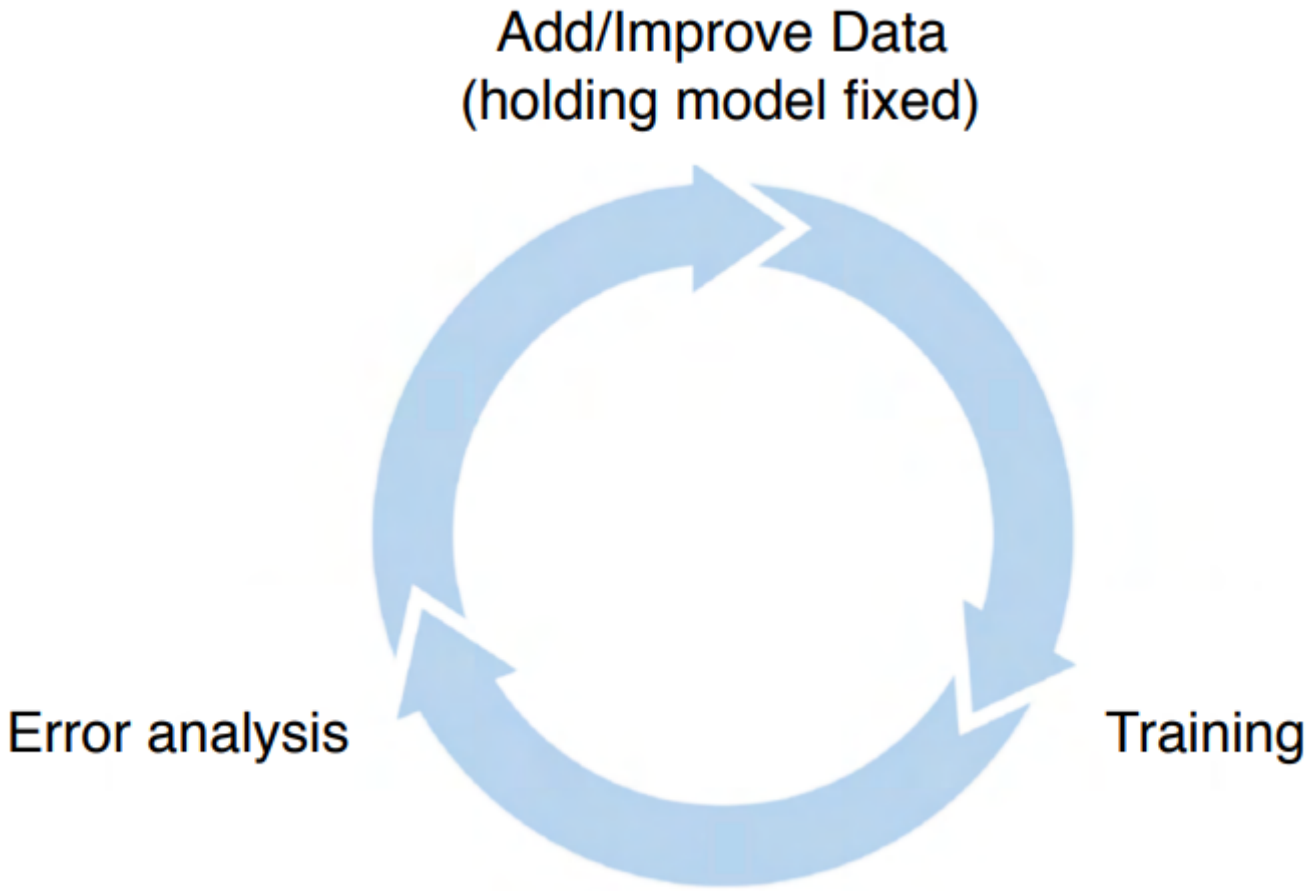
Type	Accuracy	Human level performance	Gap to HLP	% of data
Clean Speech	94%	95%	1%	60%
Car Noise	89%	93%	4%	40%
People Noise	87%	89%	2%	30%
Low Bandwidth	70%	70%	0%	6%

Data iteration : Data-centric AI development

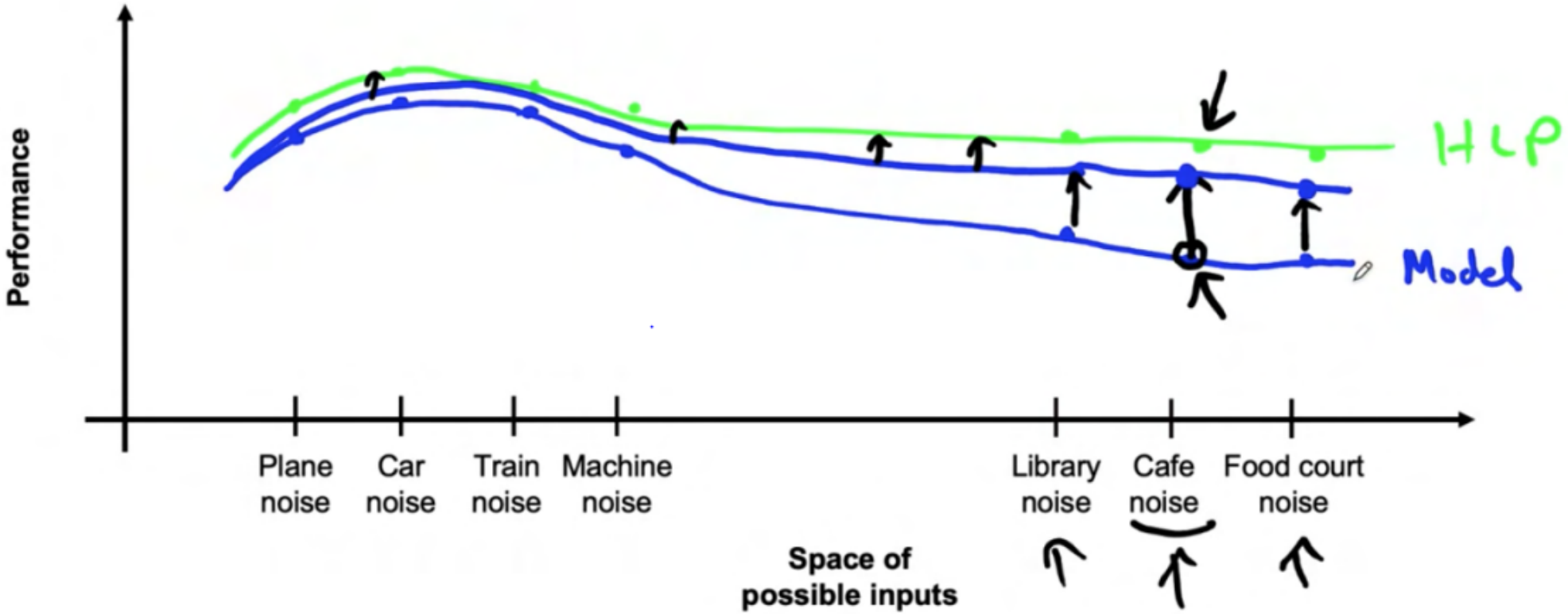
Data-centric AI development

Model-centric view	Data-centric view
Collect what data you can, and develop a model good enough to deal with the noise in the data.	The consistency of the data is paramount. Use tools to improve the data quality; this will allow multiple models to do well.
Hold the data fixed and iteratively improve the code/ model.	Hold the code fixed and iteratively improve the data.

Data iteration loop



Speech recognition example

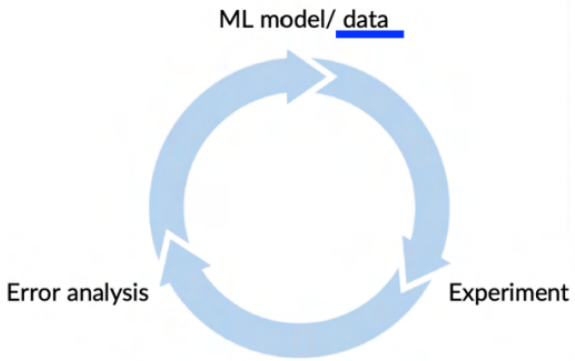


Adding data

For categories you want to prioritize:

- **Collect more data (or improve label accuracy)**
- **Use data augmentation to get more data**

Type	Accuracy	Human level performance	Gap to HLP	% of data
Clean Speech	94%	95%	1%	60%
Car Noise	89%	93%	4%	40%
People Noise	87%	89%	2%	30%
Low Bandwidth	70%	70%	0%	6%



Data iteration : Data augmentation

Data augmentation

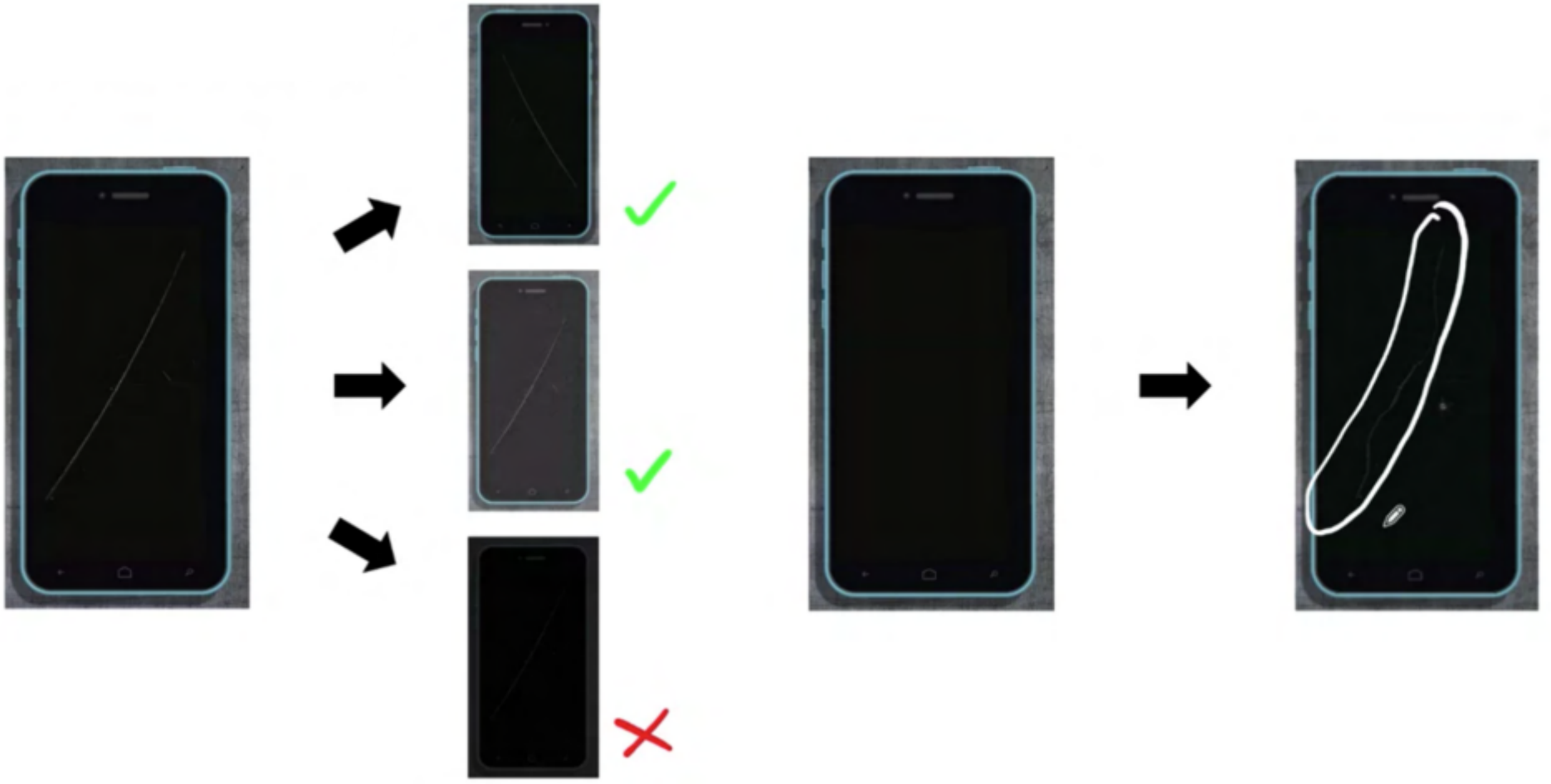
Goal:

Create realistic examples that (i) the algorithm does poorly on,but (ii) humans (or other baseline) do well on

Checklist:

- Does it sound realistic?
- Is the X $\longrightarrow$ Y mapping clear? (e.g., can humans recognize speech?)
- Is the algorithm currently doing poorly on it?

Image example



Data iteration:Can adding data hurt?

Can adding data hurt performance?

For unstructured data problems, if:

- The model is large (low bias).
- The mapping X⟶Y is clear (e.g., humans can make accurate predictions).

Then, **adding data rarely hurts accuracy.**

Photo OCR counterexample

1
High accuracy

l
Low accuracy

1? l?

Adding a lot of new “l”s may skew the dataset and hurt performance.

Data iteration : From big data to good data

From Big Data to Good Data

Try to ensure consistently high-quality data in all phases of the ML project lifecycle.

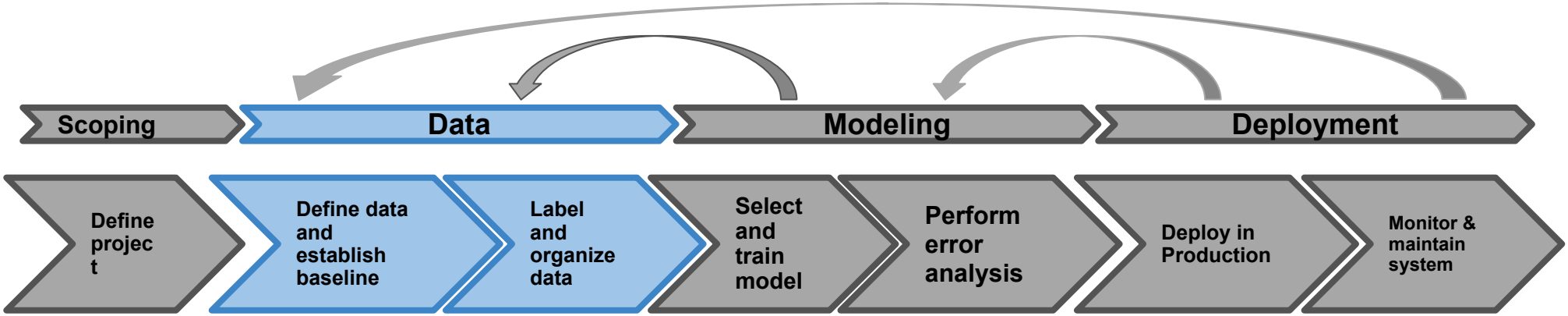
Good data is:

- Cover of important cases (good coverage of inputs x)
- Defined consistently (definition of labels y is unambiguous)
- Has timely feedback from production data (distribution covers data drift and concept drift)
- Sized appropriately

Any Questions?

Data

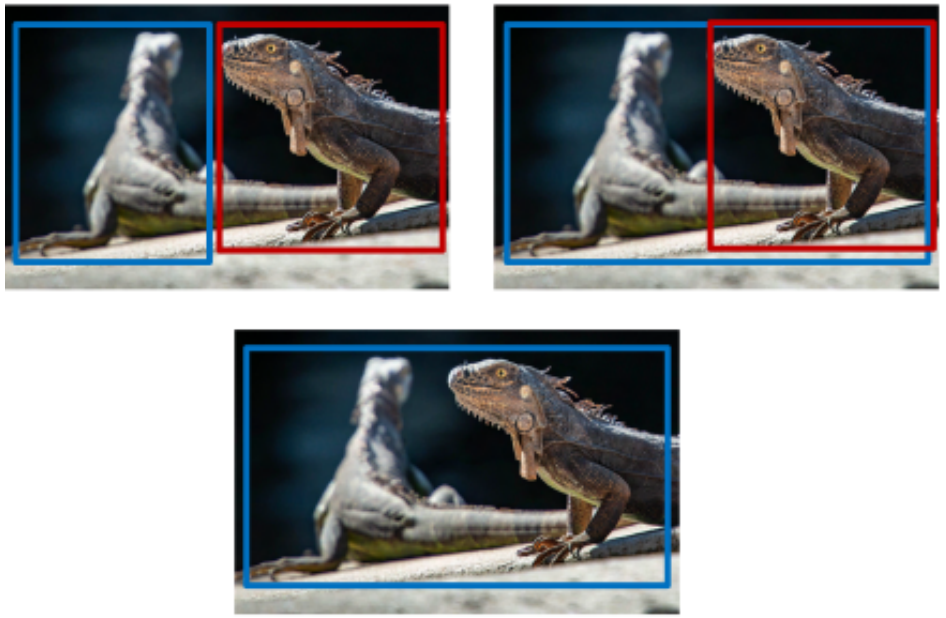
Data stage


Scoping
Data
Modeling
Deployment
Define projec t
Define data and establish baseline
Label and organize data
Select and train model
Perform error analysis
Deploy in Production
Monitor & maintain system

Define data and establish baseline

Why is data definition hard?

Iguana detection example



Labeling instructions: "Use bounding boxes to indicate the position of iguanas"

Define data and establish baseline

Phone defect detection

More label ambiguity examples

Speech recognition example

"Um, nearest gas station"
"Umm, nearest gas station"
”Nearest gas station [unintelligible]"

More label ambiguity examples

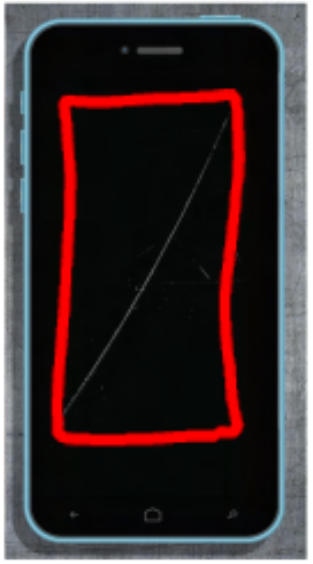
User ID merge example

	Job Board (website)	Resume chat (app)
Email	nova@deeplearning.ai	nova@chatapp.com
First Name	Nova	Nova
Last Name	Ng	Ng
Address	1234 Jane Way	?
State	CA	?
Zip	94304	94304

1 if same
0 if different

More label ambiguity examples

Data definition questions



What is the input x?

- Lightning? Contrast? Resolution?
- What features need to be included?
- What is the target label y?
- How can we ensure labelers give consistent labels?

Major types of data problems

	Unstructured	Structured
Small data	Manufacturing visual inspection from 100 training examples	Housing price prediction based on square footage, etc. from 50 training examples
Big data	Speech recognition from 50 million training examples	Online shopping recommendations for 1 million users

Clean labels are critical.

Emphasis on data process.

Humans can label data.
Data augmentation.

Harder to obtain more data.

Unstructured vs. structured data

Unstructured data

- May or may not have huge collection of unlabeled examples x.
- Humans can label more data.
- Data augmentation more likely to be helpful.

Structured data

- May be more difficult to obtain more data.
- Human labeling may not be possible (with some exceptions).

Small data vs. big data

Small data

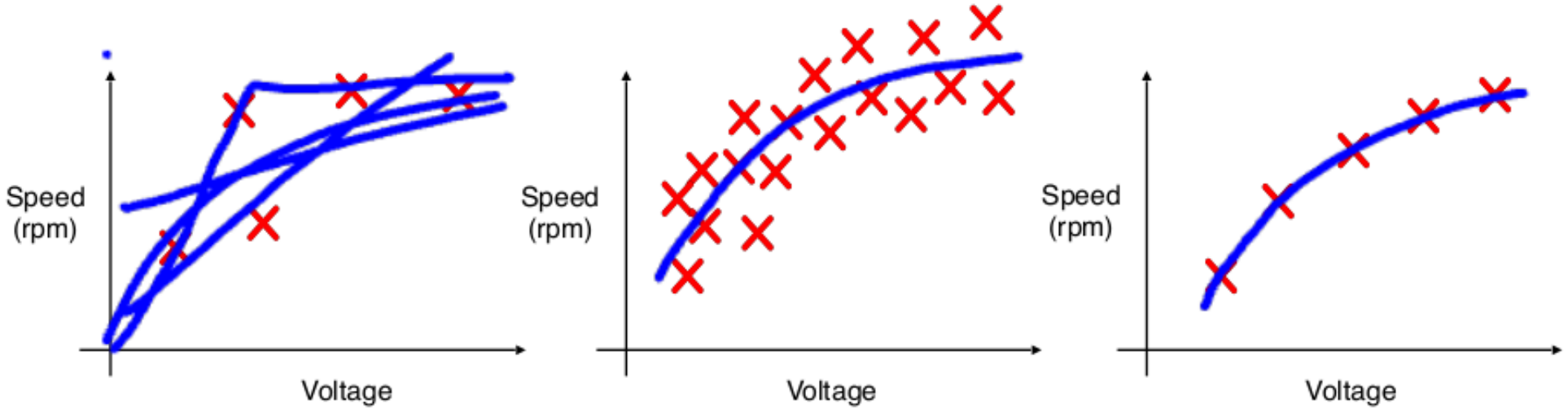
- Clean labels are critical.
- Can manually look through dataset and fix labels.
- Can get all the labelers to talk to each other.

Big data

- Emphasis data process.

Small data and label consistency

Why label consistency is important

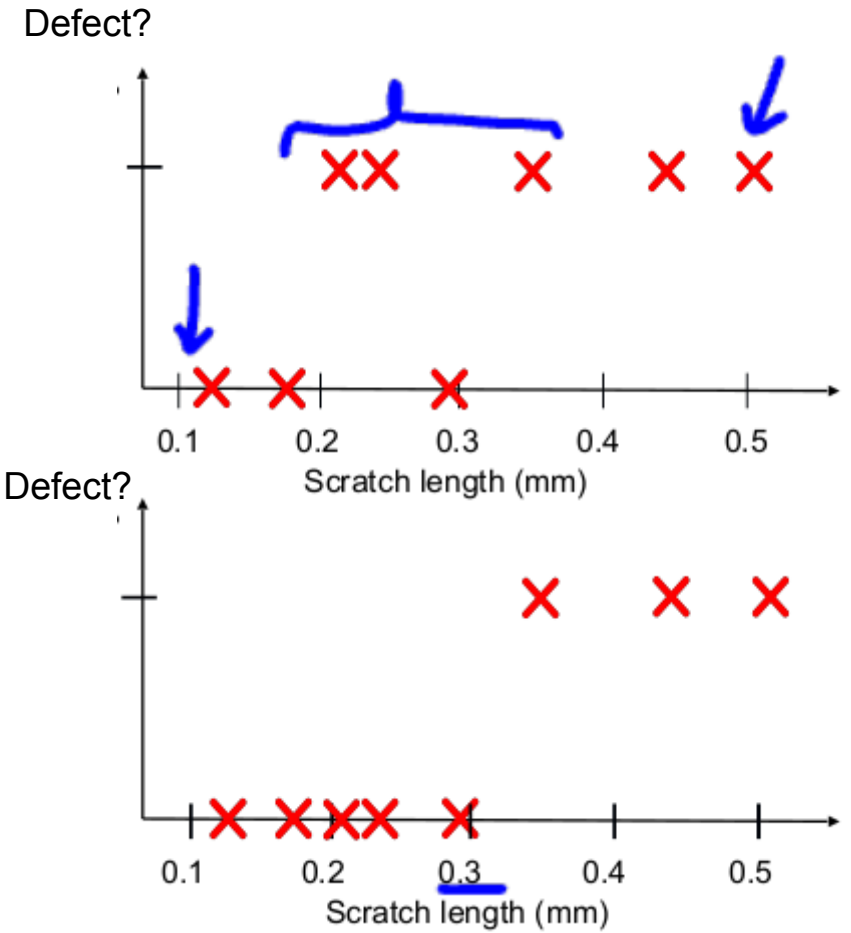


- Small data
- Noisy labels

- Big data
- Noisy labels

- Small data
- Clean (consistent) labels

Phone defect example



Big data problems

Big data problems can have small data challenges too

Problems with a large dataset but where there's a long tail of rare events in the input will have small data challenges too.

- **Web search**
- **Self-driving cars**
- **Product recommendation systems**

Improving label consistency

Have multiple labelers label same example.

- When there is disagreement, have MLE, subject matter expert (SME) and/or labelers discuss definition of y to reach agreement.
- If labelers believe that x doesn't contain enough information, consider changing x.
- Iterate until it is hard to significantly increase agreement.

Examples

- **Standardize labels**

"Um, nearest gas station"
"Umm, nearest gas station"
”Nearest gas station [unintelligible]”

"Um, nearest gas station"

- **Merge classes**

Deep scratch

Shallow scratch

Scratch

Have a class/label to capture uncertainty

- **Defect: 0 or 1**

Alternative: 0, Borderline, 1

Unintelligible audio

“nearest go”
“nearest grocery”

“nearest [unintelligible]”

Small data vs. big data (unstructured data)

Small data

! Usually small number of labelers.

! Can ask labelers to discuss specific labels.

Big data

! Get to consistent definition with a small group.

! Then send labeling instructions to labelers.

! Can consider having multiple labelers label every example and using voting or consensus labels to increase accuracy.

Human level performance (HLP)

Why measure HLP?

Estimate Bayes error / irreducible error to help with error analysis and prioritization.

Ground Truth Label	Inspector
1	1
1	0
1	1
0	0
0	0
0	1

Other uses of HLP

In academia, establish and beat a respectable benchmark to support publication.

- Business or product owner asks for 99% accuracy. HLP helps establish a more reasonable target.
- “Prove” the ML system is superior to humans doing the job and thus the business or product owner should adopt it.

The problem with beating HLP as a “proof” of ML “superiority”

"Um... nearest gas station"

"Um, nearest gas station"

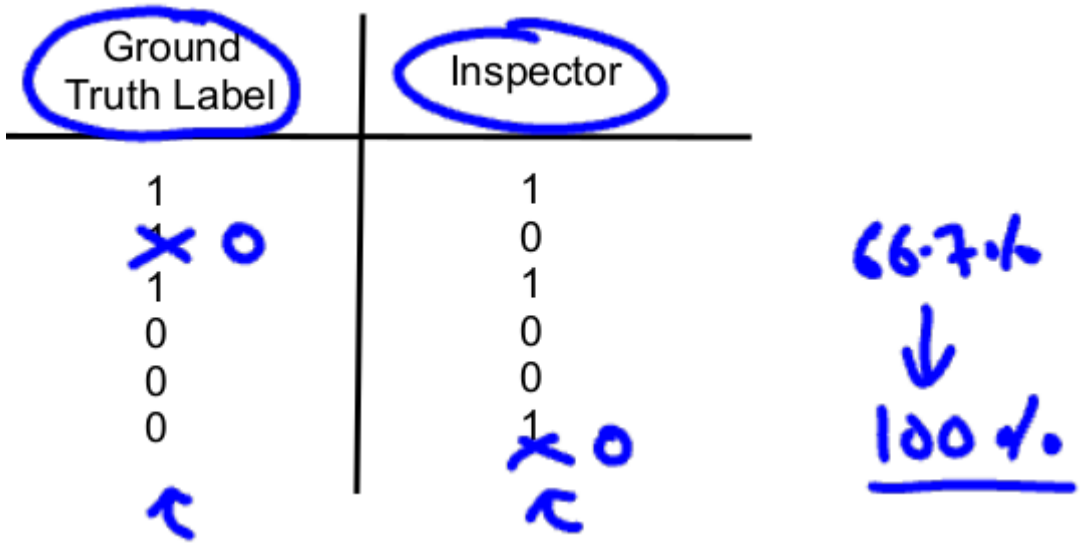
Two random labelers agree:

ML agrees with humans:

The 12% better performance is not important for anything! This can also mask more significant errors ML may be making.

Raising HLP

When the ground truth label is externally defined, HLP gives an estimate for Bayes error / irreducible error. But often ground truth is just another human label.



Raising HLP

- When the label y comes from a human label, HLP << 100% may indicate ambiguous labeling instructions.
- Improving label consistency will raise HLP.
- This makes it harder for ML to beat HLP. But the more consistent labels will raise ML performance, which is ultimately likely to benefit the actual application performance.

HLP on structured data

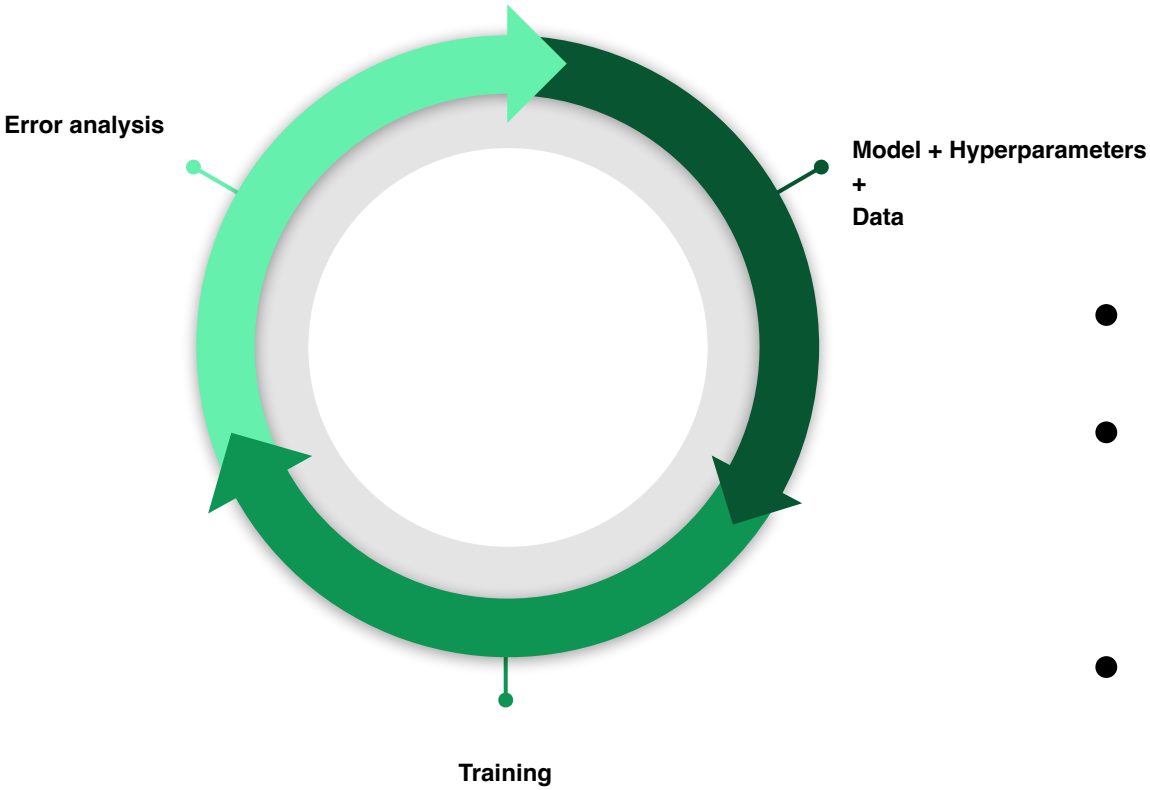
Structured data problems are less likely to involve human labelers, thus HLP is less frequently used.

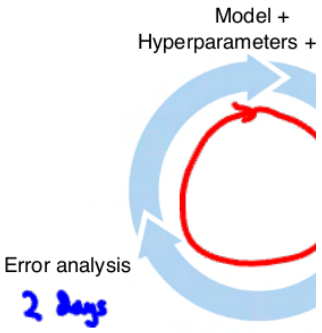
Some exceptions:

- **User ID merging: Same person?**
- **Based on network traffic, is the computer hacked?**
- **Is the transaction fraudulent?**
- **Spam account? Bot?**
- **From GPS, what is the mode of transportation – on foot, bike, car, bus?**

Label and organize data (Obtaining data)

How long should you spend obtaining data?





- Get into this iteration loop as quickly possible.
- Instead of asking: How long it would take to obtain m examples? Ask: How much data can we obtain in k days.
- Exception: If you have worked on the problem before and from experience you know you need m examples.

Inventory data

Brainstorm list of data sources (speech recognition)

Source	Amount	Cost
Owned	100h	$0
Crowdsourced – Reading	1000h	$10000
Pay for labels	100h	$6000
Purchase data	1000h	$10000

Other factors: Data quality, privacy, regulatory constraints

Labeling data

- Options: In-house vs. outsourced vs. crowdsourced
- Having MLEs label data is expensive. But doing this for just a few days is usually fine.
- Who is qualified to label?
  - Speech recognition – any reasonably fluent speaker
  - Factory inspection, medical image diagnosis – SME (subject matter expert)
  - Recommender systems – maybe impossible to label well
  - Don’t increase data by more than 10x at a time

Data pipeline

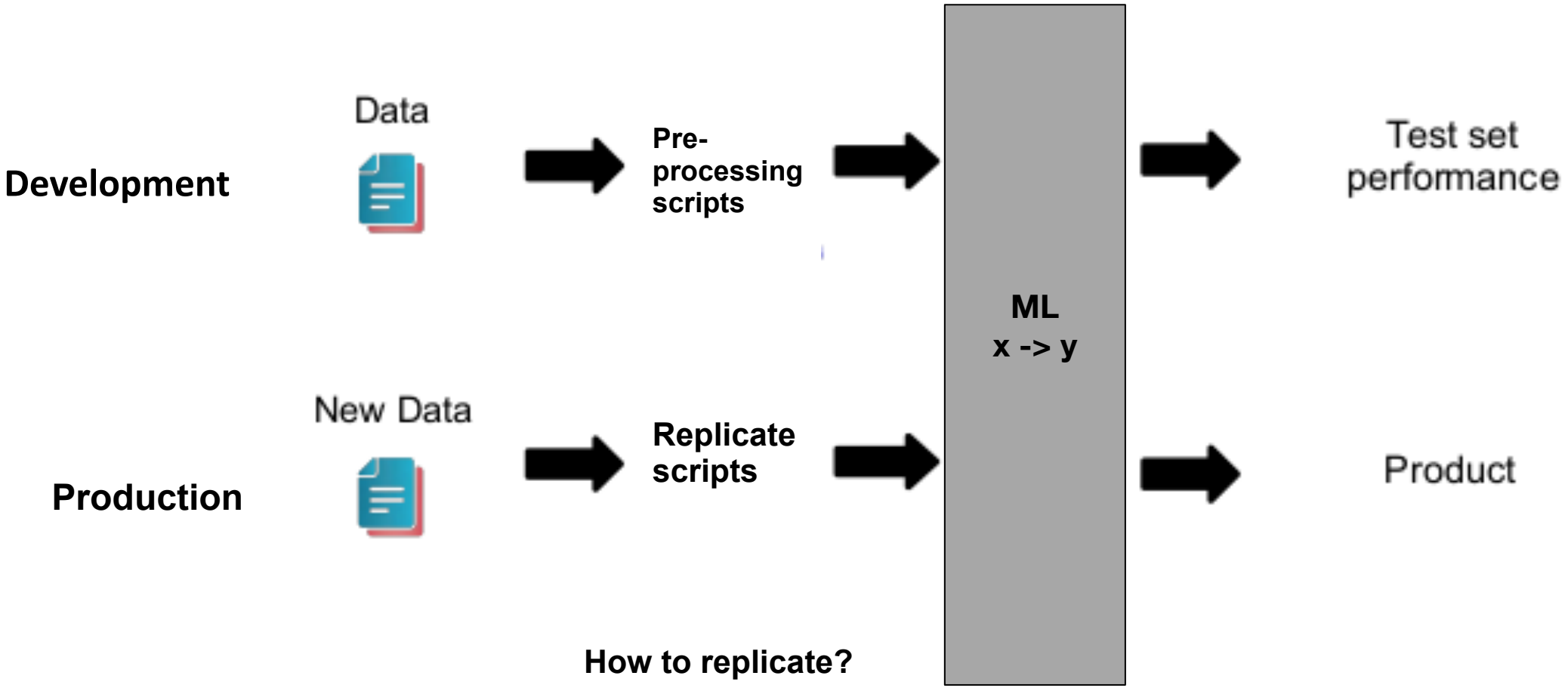
Data pipeline example

	Job Board (website)	Resume chat (app)
Email	nova@deeplearning.ai	nova@chatapp.com
First Name	Nova	Nova
Last Name	Ng	Ng
Address	1234 Jane Way	?
State	CA	?
Zip	94304	94304

x = user info

y = looking for job

Data pipeline example



POC and Production phases

POC (proof-of-concept):

- Goal is to decide if the application is workable and worth deploying.
- Focus on getting the prototype to work!
- It's ok if data pre-processing is manual. But take extensive notes/comments.

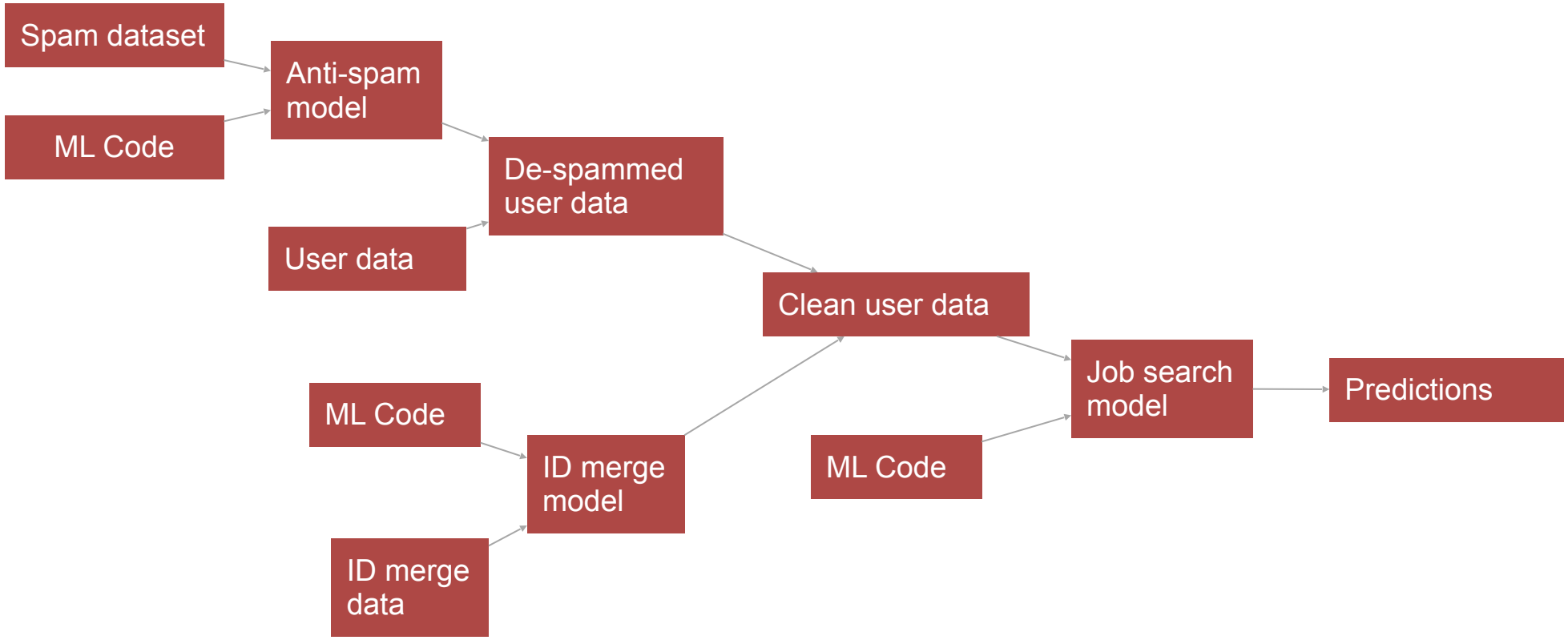
Production phase:

- After project utility is established, use more sophisticated tools to make sure the data pipeline is replicable.
  - E.g., TensorFlow Transform, Apache Beam, Airflow,….

Meta-data, data provenance and lineage

Data pipeline example

Task: Predict if someone is looking for a job. (x = user data, y = looking for a job?)



Keep track of data provenance and lineage

where it comes from

sequence of steps

Meta-data

Examples:

Manufacturing visual inspection: Time, factory, line #, camera settings, phone model, inspector ID,….

Speech recognition: Device type, labeler ID, VAD model ID,….

Useful for:

- **Error analysis. Spotting unexpected effects.**
- **Keeping track of data provenance.**

Balanced train/dev/test splits

Balanced train/dev/test splits in small data problems

Visual inspection example: 100 examples, 30 positive (defective)

Train/dev/test:

Random split:

Want:

No need to worry about this with large datasets – a random split will be representative.

Any Questions?

Scoping

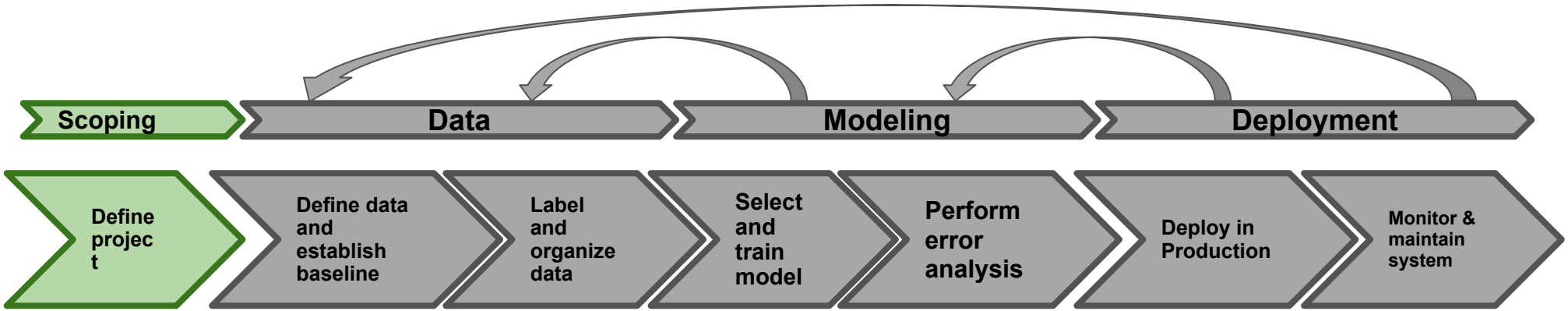
What is scoping?

Scoping example: Ecommerce retailer looking to increase sales

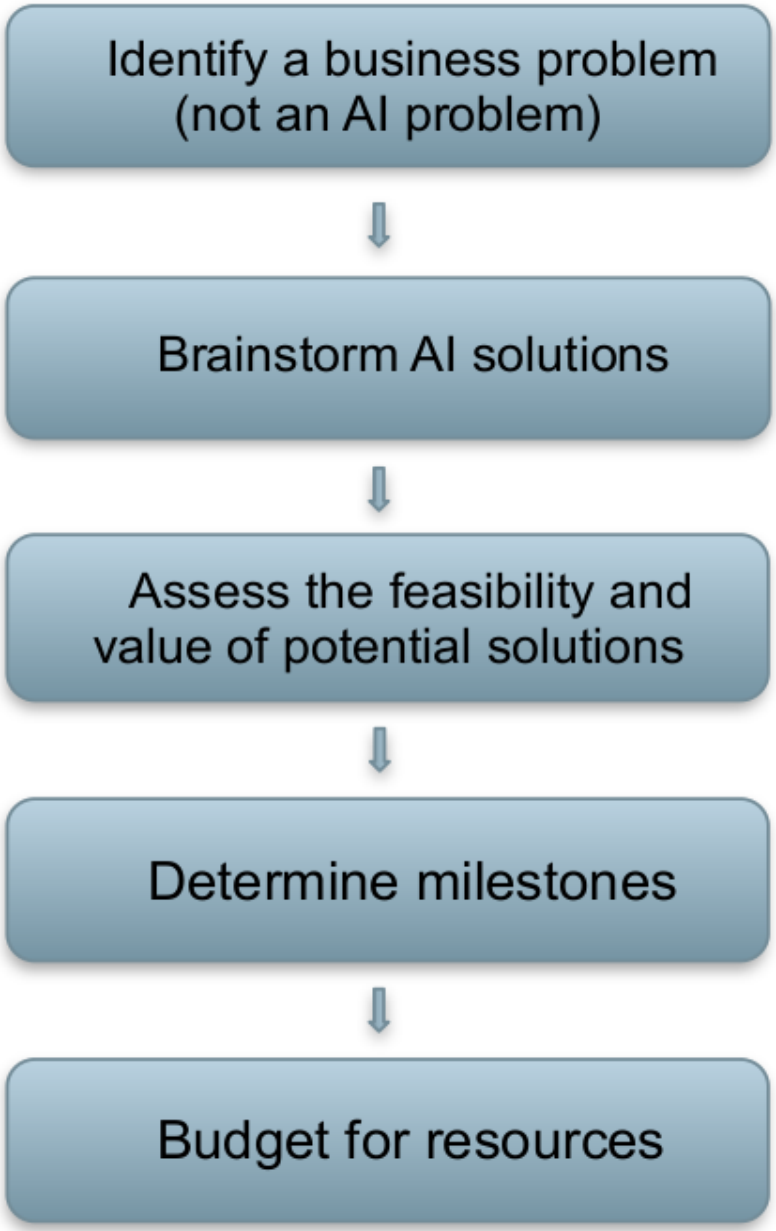
- Better recommender system
- Better search
- Improve catalog data
- Inventory management
- Price optimization

Questions:

- What projects should we work on?
- What are the metrics for success?
- What are the resources (data, time, people) needed?



Scoping process



What are the top 3 things you wish were working better?

- **Increase conversion**
- **Reduce inventory**
- **Increase margin (profit per item)**

Separating problem identification from solution

Problem	Solution
Increase conversion	Search, recommendations
Reduce inventory	Demand prediction, marketing
Increase margin (profit per item)	Optimizing what to sell (e.g., merchandising), recommend bundles
What to achieve	How to achieve

Diligence on feasibility and value

Feasibility: Is this project technically feasible?

Use external benchmark (literature, other company, competitor)

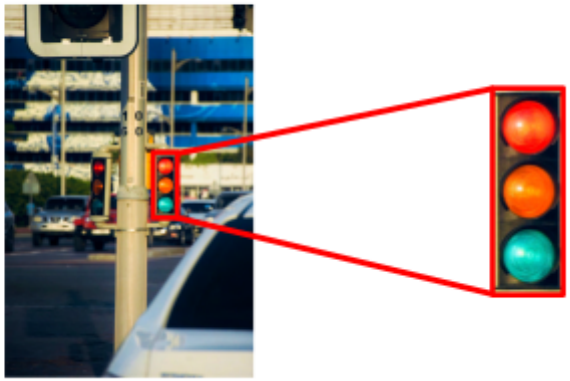
	Unstructured (e.g., speech, images)	Structured (e.g., transactions, records)
New	HLP	Predictive feature available?
Existing	HLP History of project	New Predictive features? History of project

HLP: Can a human, given the same data, perform the task?

Why use HLP to benchmark?

People are very good on unstructured data tasks

Criteria: Can a human, given the same data, perform the task?



Do we have features that are predictive?

Given pas purchase, predict future purchases

Given past purchases, predict future purchases

Given weather, predict shopping mall foot traffic

Given DNA info, predict heart disease

Given social media chatter, predict demand for a clothing style

Given history of a stock price, predict future price of that stock

Diligence on value

MLE metrics				Business metrics
Word-level accuracy	Query-level accuracy	search result quality	User engagement	Revenue

Have technical and business teams try to agree on metrics that both are comfortable with.

Ethical considerations

- **Is this project creating net positive societal value?**
- **Is this project reasonably fair and free from bias?**
- **Have any ethical concerns been openly aired and debated?**

Milestones and resourcing

Milestones

Key specifications:

- **ML metrics (accuracy, precision/recall, etc.)**
- **Software metrics (latency, throughput, etc. given compute resources)**
- **Business metrics (revenue, etc.)**
- **Resources needed (data, personnel, help from other teams)**

Timeline:

- **If unsure, consider benchmarking to other projects, or building a POC (Proof of Concept) first.**

Any Questions?