Wadhwani AI

www.wadhwaniai.org

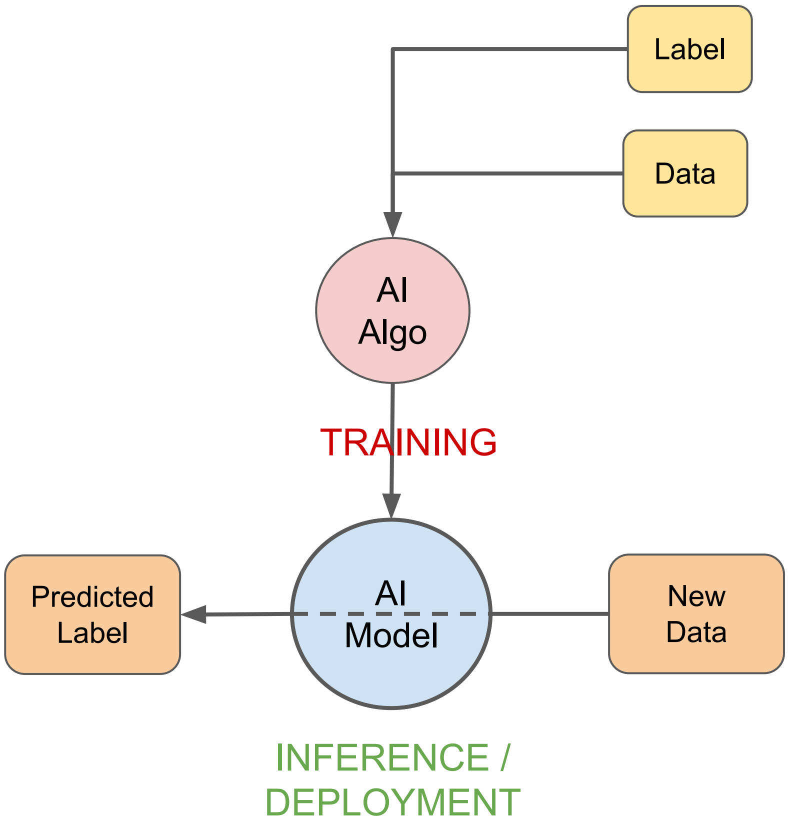
Applied AI: data issues, challenges, paths to success

Alpan Raval, Ph.D.
Chief Scientist, AI/ML

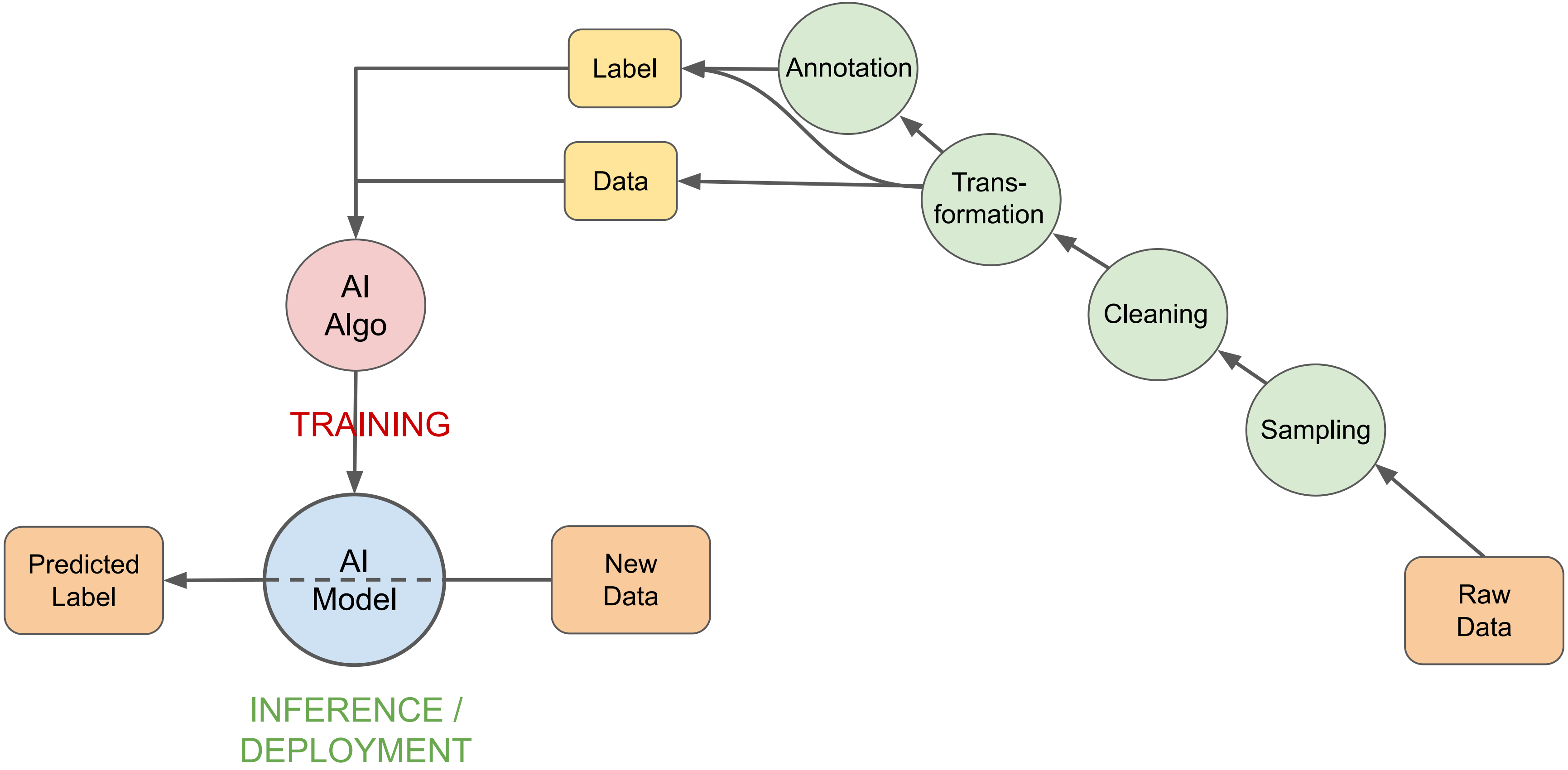
Outline

- How AI works: the central role of data
- Data concepts and challenges
- Case studies
  - Integrated pest management through AI
  - Early prediction of patient risk for dropping off medication regimen
- Successful deployment of AI models: ingredients

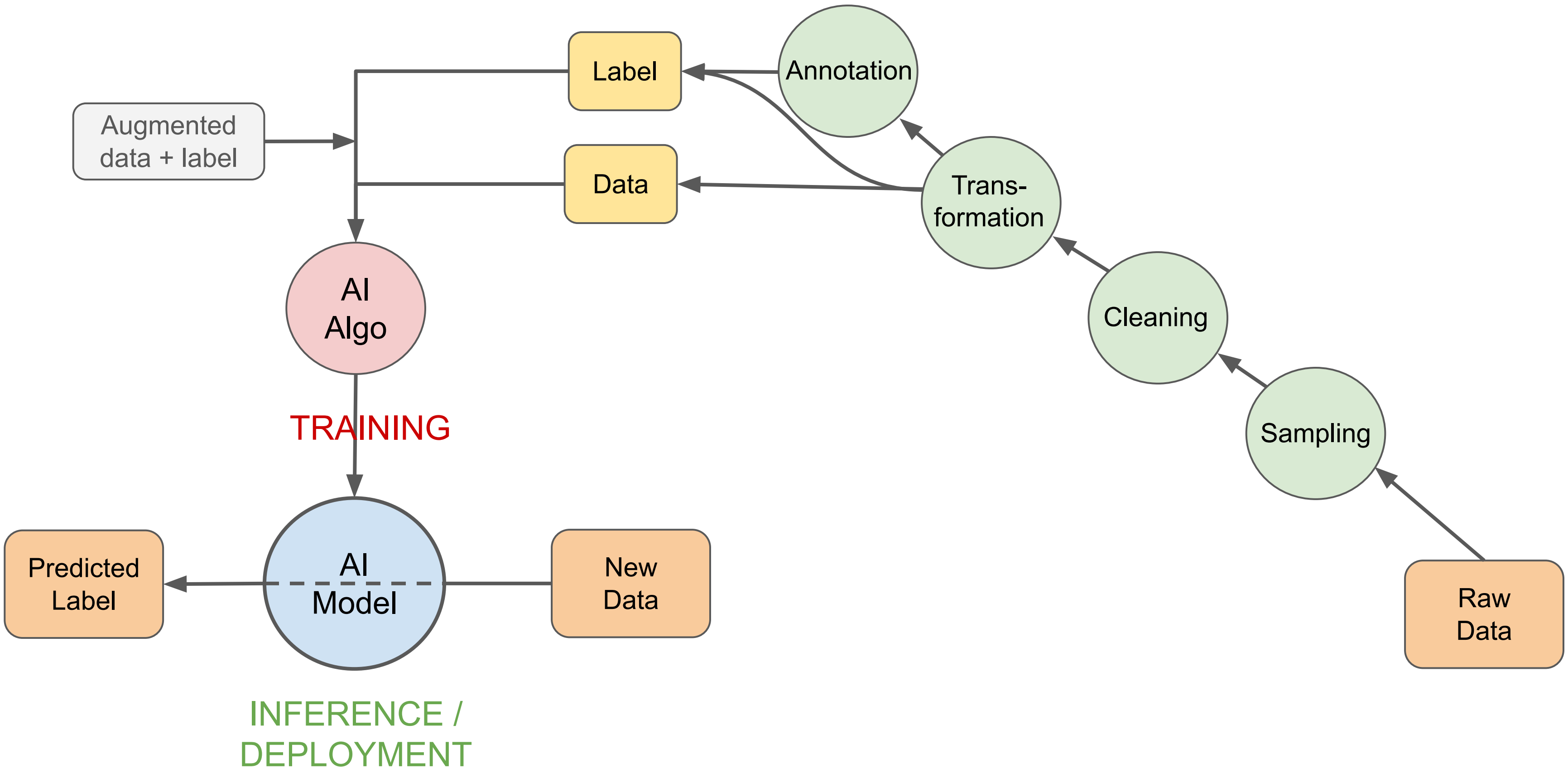
How AI works


Label
Data
AI
Algo
TRAINING
Predicted
Label
AI
Model
New
Data
INFERENCE /
DEPLOYMENT

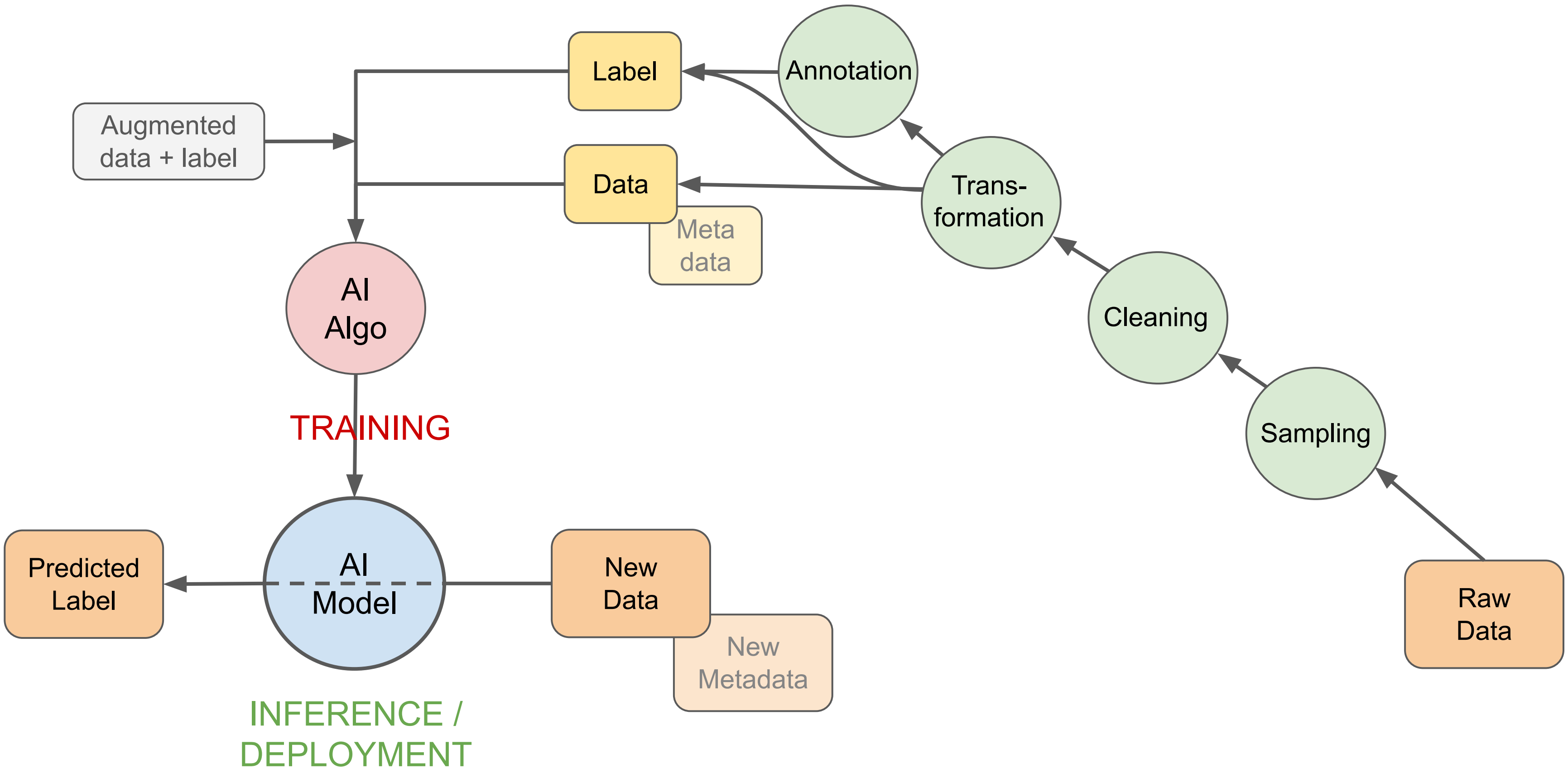
How AI works


Label
Annotation
Data
Trans-
formation
AI
Algo
Cleaning
Sampling
TRAINING
Predicted
Label
AI
Model
New
Data
Raw
Data
INFERENCE /
DEPLOYMENT

How AI works

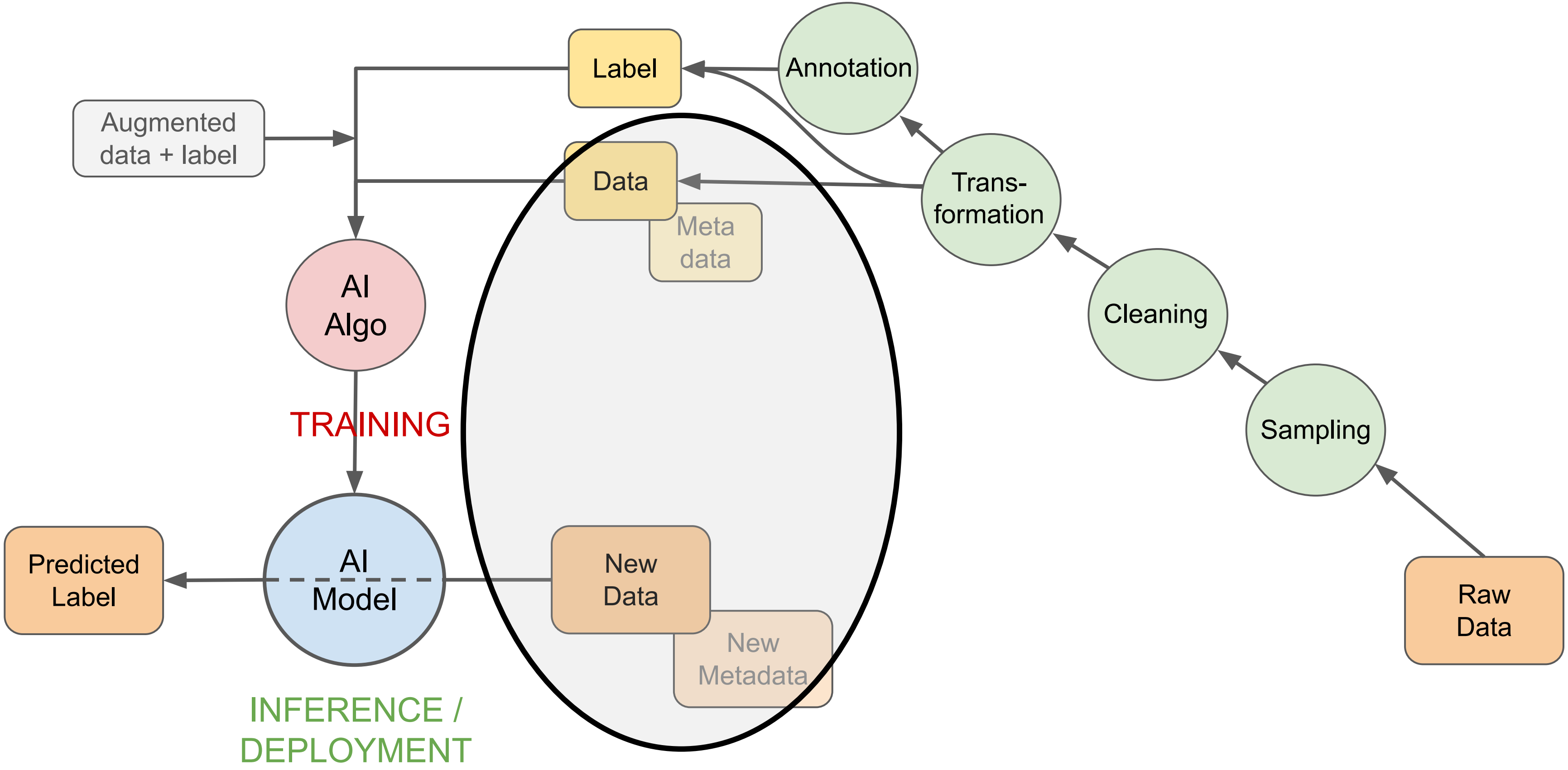

Label
Annotation
Augmented data + label
Data
Trans-formation
AI Algo
Cleaning
TRAINING
Sampling
Predicted Label
AI Model
New Data
Raw Data
INFERENCE / DEPLOYMENT

How AI works

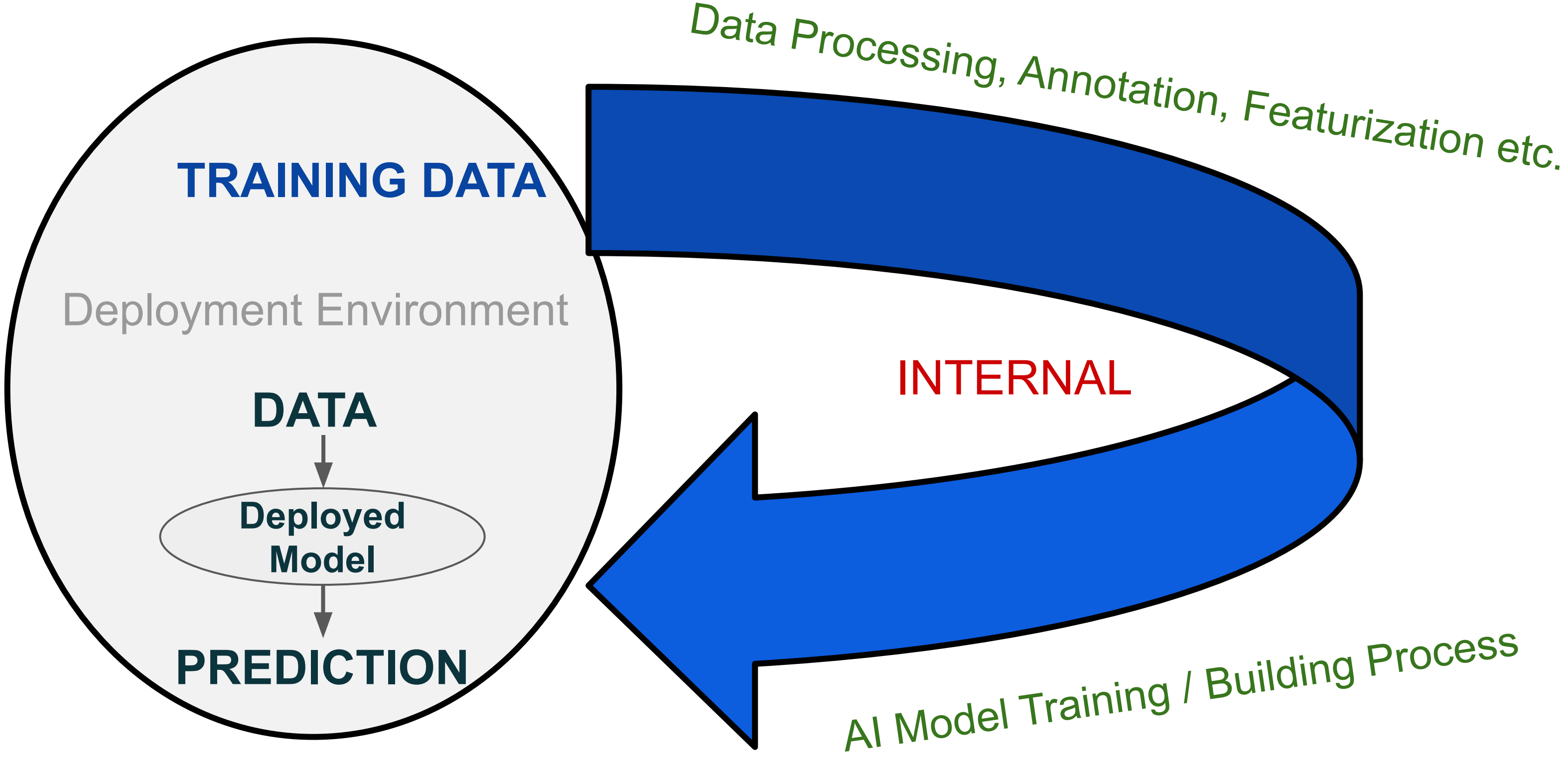


Label
Annotation
Augmented data + label
Data
Trans-formation
Meta data
AI Algo
Cleaning
TRAINING
Sampling
Predicted Label
AI Model
New Data
New Metadata
Raw Data
INFERENCE / DEPLOYMENT

How AI works

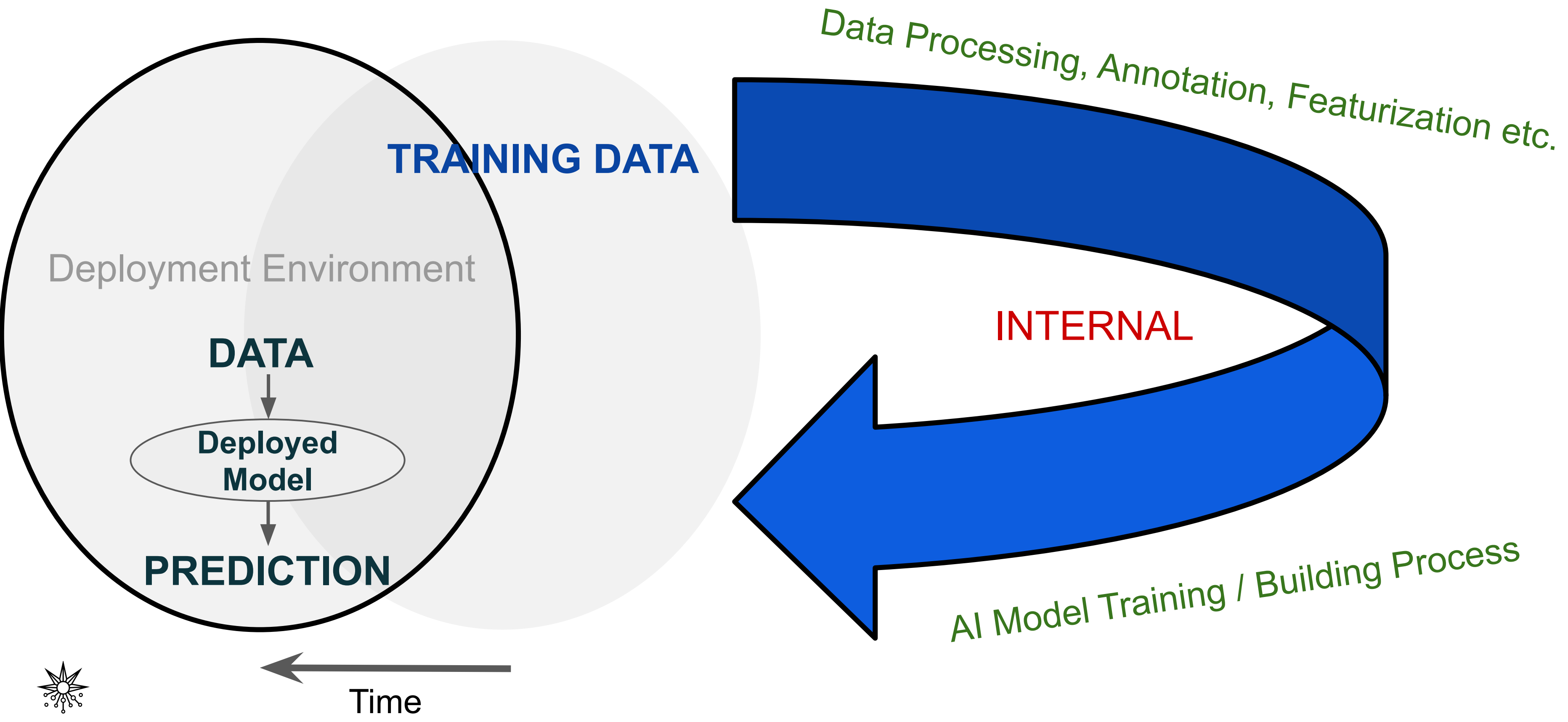

Label
Annotation
Augmented data + label
Data
Trans-formation
Meta data
AI Algo
Cleaning
TRAINING
Sampling
Predicted Label
AI Model
New Data
Raw Data
New Metadata
INFERENCE / DEPLOYMENT

Deployment and Data



Training and Inference Data should ideally come from the same environment! Few exceptions …

Data Drift


Data Processing, Annotation, Featurization etc.
TRAINING DATA
Deployment Environment
DATA
Deployed Model
PREDICTION
INTERNAL
AI Model Training / Building Process
Time

Data Basics 1

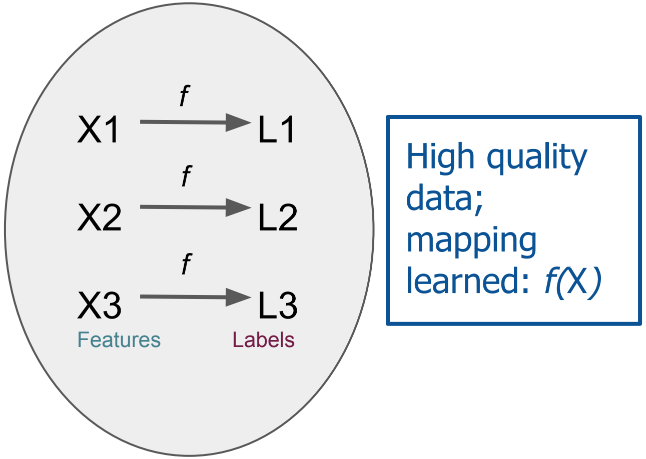
- **Training data** is data used for the AI model to **learn** mapping between **covariates / features** and **outcomes / labels**. Often, creation of the label or outcome part of the training data requires **annotation**.
  - Examples:
    - (patient covariates (features), patient outcome (label))
    - (baby video (features), baby weight (label))
    - (pest trap image (features), pest count (label))
- Collection of **training data** is an internal exercise, which involves an algorithm that is run on the training data. The output of this exercise is an **AI model**, which is a software file containing numerical values of various parameters + instructions (code) for applying these parameters to new data (covariates/features), thus **predicting a label** on the new data.
- A (supervised) **AI model cannot be developed without training data**.
- **Inference data** is the data on which the AI model makes a prediction. It does not require a label since the point of the AI model is to predict a label for it.
- Model performance is **very sensitive** to training data AND inference data!

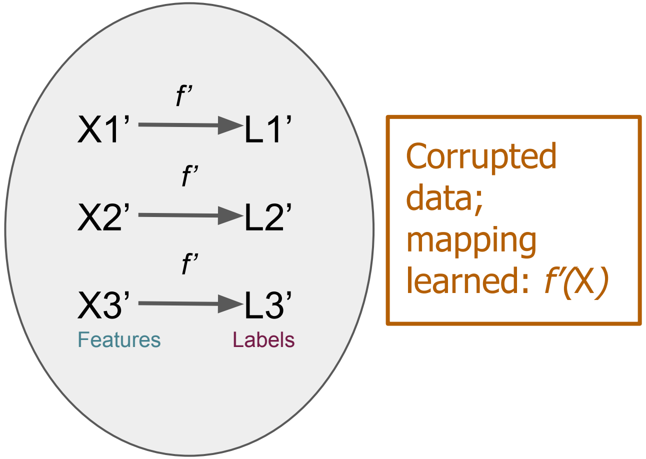
Data Basics 2

- As part of the internal AI model development process, **Training data** is often split into a **training set**, a **validation set**, and a **test set**. This is called a **data split**.
  - **Training set**: data used to train the model, or learn mapping between features and labels.
  - **Validation (val) set**: data used to find the best values of the hyperparameters of the model (hyperparameter optimization).
    - Hyperparameter examples: number of training cycles, learning rate, etc.
  - **Test set**: data that is (finally) used to evaluate the model internally.
- The data split is sometimes done in a **random** manner, sometimes in a **longitudinal** manner. Longitudinal splits are especially useful when you expect data characteristics to change over time. Other types of splits are also possible, depending on how one wants to evaluate the model internally.
- Once we are satisfied that the AI model has done well on the test set, it is then **deployed**, used in the real world. At that point there is no more training or learning. The model is exposed to real, inference data, and makes label predictions on that data.

Data Quality

- **What is data quality?**
  - Having accurate and complete features and having accurate and complete labels.
- **Why is data quality important?**
  - AI algorithm has to learn how to map features to labels. If it learns the wrong mapping or a random mapping, the AI model will not work well in practice.
- **If the deployment environment contains data of poor quality, should the model also not be trained on poor quality data?**
  - No. Poor quality data typically contains a lot of random errors. A model trained on data with random errors will learn a random mapping. It will not perform better on poor quality data than a model that is trained on high quality data.





How much data is enough?

- **The more the better!** No standard sample size formulae in AI/ML.
- But there are some **ground rules** and **thumb rules**.
    - For deep learning classification models on images trained from scratch, typically need **1,000 - 2,000 images per class** (this reduces significantly with pre-trained models).
    - **Classical ML models** are less data hungry: **hundreds of rows per class** may be enough.
    - **Small, high quality datasets are better than large, low quality** datasets (sometimes no choice!).
    - **Data diversity** is important: data from different locations, time points, different conditions (all relevant to deployment). This is so that the AI algo can learn to map better.
    - **Continuous data collection** is better than a one-time dump (often this is taken care of through the deployment and evaluation process).
    - **The less accurate the data, the more data you will need**. This is because the algo needs to learn to average out errors.

Sampling, problem

- Even with good, clean data, the following scenarios can arise:
  - The training data is not **representative**. It misses certain **cohorts** that are present in the deployment environment, e.g., old people, specific locations, women, low-income groups, etc. In such a case, an AI model trained on the data will not work well for those specific cohorts.
  - One of the classes, called the **minority class**, is severely under-represented in the data, even if the total dataset is large. The AI model does not have enough samples from that class to learn mapping.
  - Different classes are under-represented in different cohorts, e.g, too few TB negative patients among migrant workers, too few TB-positive patients among women. Will result in a model that makes **biased predictions**.

Sampling, solution

- Solution to minority class issue:
  - Collect **disproportionately large number of samples from minority class** (better), more than what is present in the real world, for training the AI model.
  - **Oversample** the minority class (not as good), with replacement, during training.
  - Oversampling / disproportionate collection may have to be done **separately on different cohorts** if there is a different minority class for each cohort.
  - In either case, the AI model should usually be **evaluated on data that represents proportions found in the real world**, since that is where the model will eventually be deployed.

Annotation 1

- **Annotation** is the process of a human assigning a label to data. Sometimes the label is part of the data collection effort.
- The AI algo learns the mapping between features and the annotated label. For new data, it has then learned to predict the label. Verifying the predicted label through human annotation is the process of **evaluation** of the AI model.
- **Examples**: person has / does not have disease; bounding boxes on abnormal features in X-rays; bounding boxes around pests; image is good / poor quality.
- Sometimes annotation is required for internal ML algo purposes; the annotation, in such cases, is not the label that the algo will try to predict. Example: keypoint annotation in anthro.
- **High quality annotation** is an essential component of high quality training data.
- When good annotation is difficult and requires experts, it is often useful to create a **gold standard dataset** that is annotated well, by experts. This dataset is usually used for evaluating models, not training them.

Annotation 2

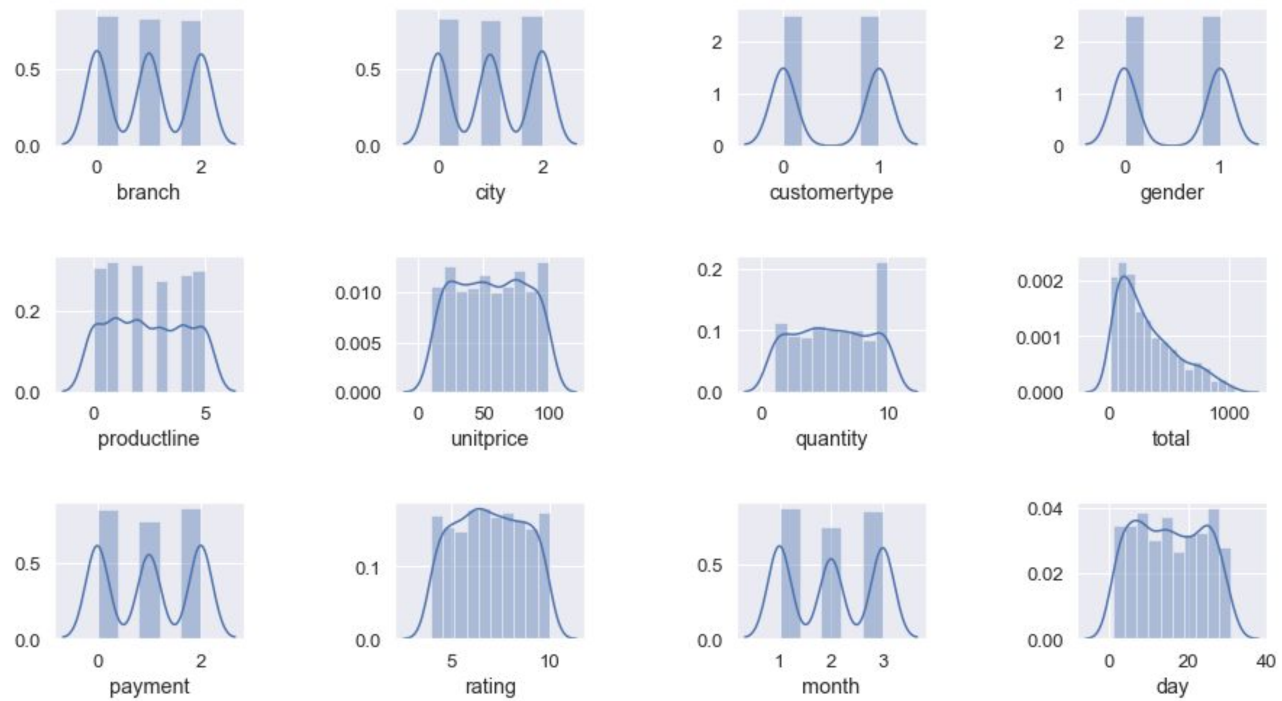
- Sometimes, **multiple annotation** is necessary on the same data.
- The accuracy or difficulty of an annotation task is measured by various measures of **inter-annotator agreement**. Note that this is different from the accuracy of the AI model.
- Multiple annotations can be incorporated into the training of an AI model, either by **aggregating** the annotations in some way or changing how the **optimization** problem in the AI algo is solved.
- The goal of good annotation is to **improve inter-annotator agreement**, often by codifying (in writing) the process of annotation. Note that this has to be done in such a manner that the annotation still makes sense in a real-world context, since the result of the annotation is what the AI model will try to predict in deployment.
- A (supervised) **AI model cannot be more accurate than the annotation used to train it**.

Metadata

- Metadata is, quite simply, **data about data** (what, when, where, who, how, which, why). Examples: timestamps and location info on images, collector and annotator information, cohort information, app version, …
- **Why is metadata useful?**
  - Indexing and organization of data
  - Monitoring of data quality
  - Additional features for training the AI model
  - Diagnosis and debugging of model inaccuracies. Example: is the model inaccurate because annotator X created the labels, or because the training data is too old?
  - Creation of evaluation cohorts for the AI model. Example: longitudinal splits, user gender, demographics etc.

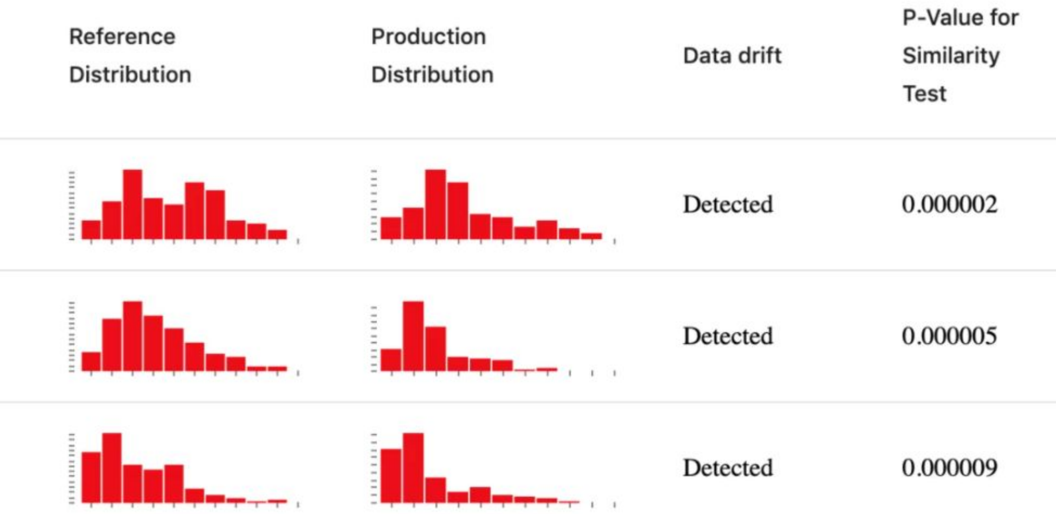
Data Distributions

- A data distribution is the probability distribution of features (or labels).
- Data distributions reveal label skews, minority classes, biases in the data that may cause problems in the resulting AI model.



Data Drift

- Data drift is said to occur when the data distribution changes (with time, across locations, across cohorts, etc.). Data drift can occur in the features, labels or mapping (**concept drift**).
- A model trained on data after data drift has occurred will generally be different from the original model, and have different performance characteristics. The original model evaluated on drifted data will also have different performance characteristics.
- It is important to monitor data drift as that may provide a diagnosis for changes in model performance.



Reference Distribution	Production Distribution	Data drift	P-Value for Similarity Test
		Detected	0.000002
		Detected	0.000005
		Detected	0.000009

Exploratory Data Analysis (EDA)

- EDA is the process of analyzing a dataset in order to understand it. It is open-ended and **exploratory** in nature.

- **Visualizing** data distributions is part of EDA. Other aspects of EDA: analyzing **completeness / missing values**, computing **summary statistics** by column, **outlier analysis**, **categorical vs. continuous** data, computing **correlations** and **mutual information** between columns, etc.

- EDA is necessary to understand the characteristics and limitations of **your** dataset.

- EDA also provides important clues on the type of AI model that will be built and further processing that needs to be done.

- EDA should also be done on metadata, if it is available.

Unsupervised AI

- **Unsupervised AI** is a type of EDA that involves discovering **patterns and groupings** in the data, without regard to the label. A new datum can then be assigned either to a pre-discovered pattern or group, or assigned to a new group. This enables classification of the new datum.
- Unsupervised AI also involves **dimensionality reduction** in order to enable visualization of a high-dimensional dataset in a lower (typically, 2) dimensional space.
- Examples:
  - Data can be clustered by similarity of feature values (**unsupervised training**). The features of a new datum can be analyzed and the new datum can be assigned to an existing cluster (**inference**) or new cluster (**outlier analysis**). Common clustering techniques are **K-means** and **hierarchical clustering**.
  - Transformations can be applied to features to create lower dimensional representations (linear: **PCA**; non-linear: **t-SNE**) that are useful for visualizations.
- Many other unsupervised/self-supervised models: **encoder-decoder**, **word embeddings** etc.

Featurization / Feature Engineering

- **Featurization or feature engineering** is a natural extension of the EDA process. It involves **transforming or encoding features** in different ways to make it easier for the AI algo to map features to labels. Examples: one-hot encodings, target encodings, non-linear feature combinations, etc.

- Feature engineering is commonly used in the context of **classical machine learning models**, and in the context of **tabular data**. Deep learning models trained on **multimedia data** (images, video, raw text, audio) rarely require complex feature engineering.

- **Feature selection**, i.e., the process of selecting the most important features for predicting the label, is sometimes part of feature engineering. More often, it is an outcome of the AI modeling process.

Metrics

- AI model development happens through the **optimization** of certain metrics that measure **closeness between model predictions and actual labels** on the training data.
- This closeness can be measured in different ways: average accuracy, sensitivity, specificity, etc. These measures are the **metrics**.
- For mathematical reasons, metrics that are used to measure the model performance in a deployed setting are not the same as the ones that are used in internal model development.
- The key metrics that the AI model should aim to reach in a deployed setting must be identified early on in model development (**both metric and numerical value**). This is because this metric is what the AI model will be evaluated against and optimized during model development and hyperparameter optimization.
- It is common to first define an **operating point** for the model, and a desired metric value at that operating point. Examples: Specificity at 90% sensitivity, Precision at 90% sensitivity, F1 score at 50% sensitivity.
- Some metrics are independent of operating point. Useful for optimizing the model, but not very useful in practical settings. Example: Area under ROC curve (**AUC-ROC**), etc.

Post-deployment: Monitoring & Evaluation

- **M&E** is one of the most important components of applied AI. The performance of an AI model MUST be monitored and evaluated post-deployment.
- **Why?**
  - AI model is not like any other piece of software; it does not behave in a deterministic manner. Accuracy / metric value is conditional on data drift, on changing deployment environment.
- Start by **continuously evaluating the accuracy** of model, or at periodic intervals, through continuous or periodic annotation of samples.
- If accuracy dips below expectation, need to diagnose. Data drift? Deployment software bug? Internal testing bug?
- Once diagnosed, fix bug or retrain on new data as necessary before next round of deployment. This is the **iterative** nature of AI.
- Note that improving the AI algo is not always necessary; in many cases, simple retraining on new data is enough. This is called **model refresh** (as opposed to new **model version**).

Post-deployment: Human-in-the-loop

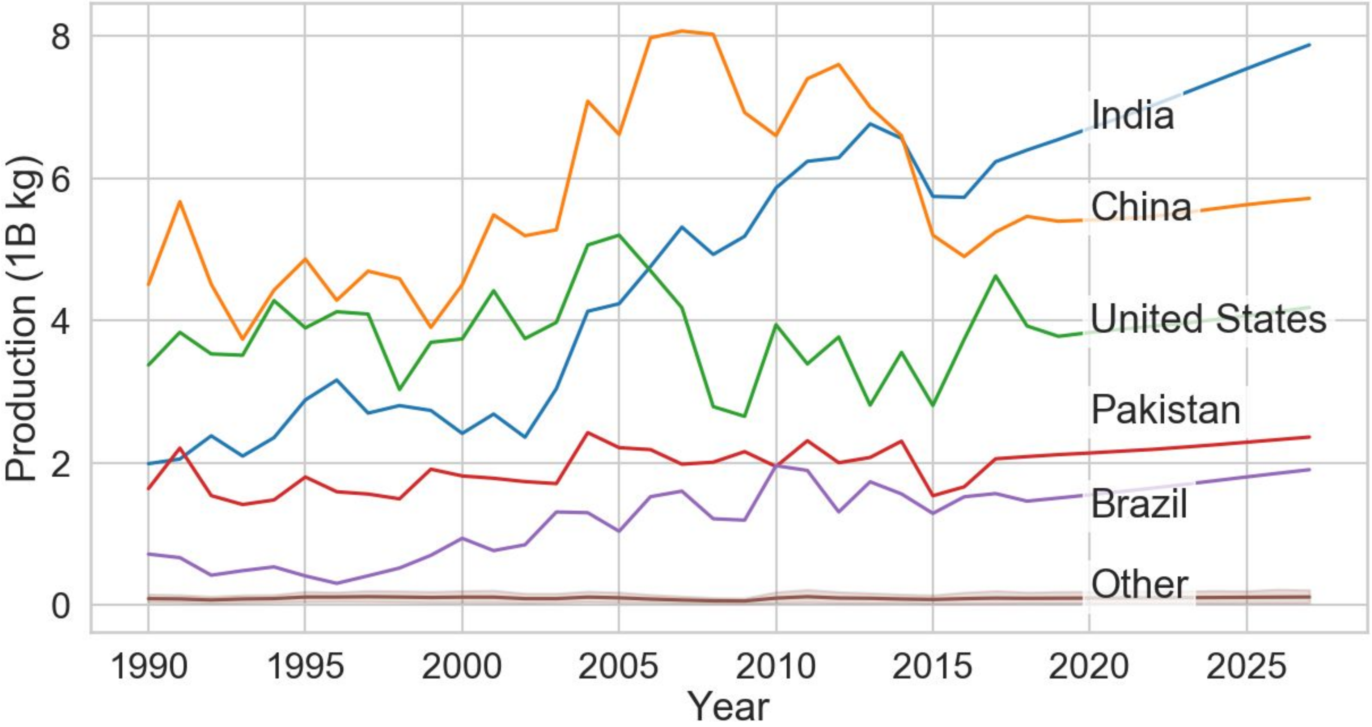
- An AI model typically does not output a label directly. It outputs a **score** that loosely measures the probability of the label taking a certain value. The **operating point** of the model involves choosing a **cutoff value** such that scores higher than the cutoff for a label imply prediction of that label.
- Higher scores imply greater confidence in the label prediction; lower scores indicate low confidence.
- The accuracy of a deployed AI model can be improved by sending data with scores close to the cutoff for human annotation, rather than believing the predicted AI label.
- The process of sending low-scoring data for human annotation is called **human-in-the-loop**. There are ways to optimize the use of human-in-the-loop. Generically, any process where a human rechecks some sample of predicted AI labels in termed as human-in-the-loop.
- When human-in-the-loop annotations are fed back to retrain the AI model in an iterative fashion, this process is called **active learning**.

AGRICULTURE CASE STUDY

Pest Management in Cotton Farms

Cotton Is A Global Cash Crop

Nearly 100 million families across the world, mostly smallholder farmers in developing countries, rely on cotton farming for their livelihood.



[Organization for Economic Cooperation and Development]

Vulnerability to Pest Infestation

Pest attacks not only cause crop loss, but push farmers already fighting poverty into despair

MUMBAI

Pink bollworm may eat up half of Maharashtra's cotton crop

Alok Deshpande

MUMBAI, NOVEMBER 17, 2017 00:34 IST
UPDATED: NOVEMBER 17, 2017 06:50 IST

SHARE ARTICLE 0 PRINT A A A

ASIA NOVEMBER 23, 2017 / 1:40 PM / 3 YEARS AGO

India cotton exports to drop as pink bollworms eat crop

3 MIN READ

Why our farmers are killing themselves

A Narayanamoorthy | P Alli | | Updated on January 27, 2018 | Published on January 26, 2017

0

As the sun sets A new dawn seems far away - SR Raghunathan

Rising input costs have shrunk profits, making cultivation unviable. Easy access to credit and better MSPs can help

Challenges and Impact of Pest Management

Cotton is the largest consumer of pesticides of all crops in the country

- **Farmers do not know when to spray**
  - Based on informal pest census
  - At regular intervals during the season
  - Upon recommendation or observation of other farmers in their area
  - Upon recommendation of pesticide dealers

- **Farmer-empowered pest management practices have been difficult to scale**

- **Pesticide usage is generally incorrect**
  - Mixing various types
  - Using too much
  - Diluting too little

- **Estimated that farmer profits can increase by 26% through smarter pest management**

Pheromone Pest Traps To Monitor Pest Populations

Traps mounted throughout the field

Lure emits pheromone to attract male bollworms

Removable top to change lure; bottom opens to release catch

Sticky traps for other pests that fly and hop

Extension Program Workflow

<table>
<tr><th>PROGRAM</th><th>DATA COLLECTION</th><th>AGGREGATION</th><th>PROCESSING</th><th>DISSEMINATION</th></tr>
<tr><td>GOVERNMENT</td><td rowspan="2">Weekly monitoring of demo plots
Manual counting of pest density</td><td>Uploaded using CropsApp</td><td>Manual analysis at an agricultural science institute</td><td>Block-level advisory through bulk SMS and field visits</td></tr>
<tr><td>WELSPUN</td><td>Uploaded using SourceTrace</td><td>Manual analysis by core team of experts</td><td>Advisory through field visits</td></tr>
</table>

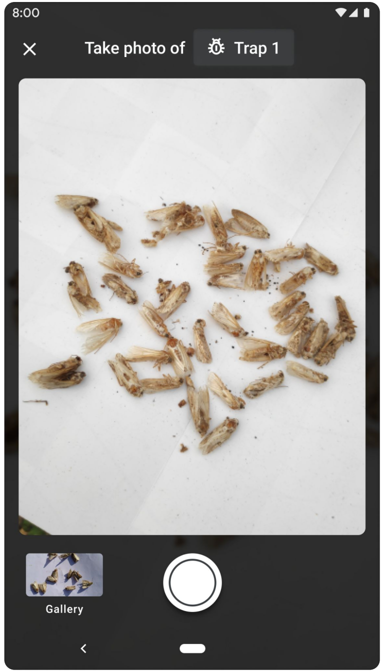
Welspun works with the Better Cotton Initiative (BCI) on app usage for early pest detection in cotton farms

Slide Source: [WIAI Cotton PM app Deck brief Eng]

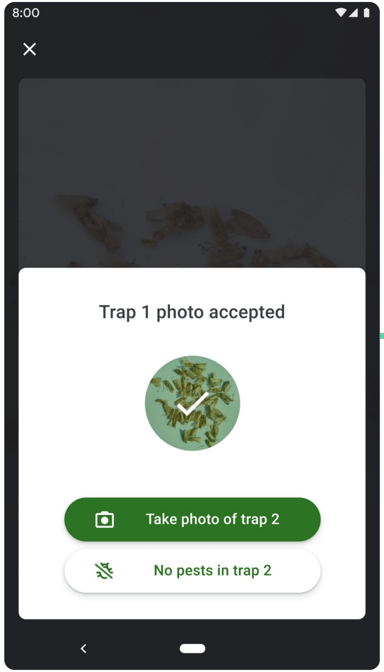
A new solution: AI-based Pest Management App

A farmer or extension worker empties the moths from the pheromone trap.

He then opens the app and captures 2 trap images.

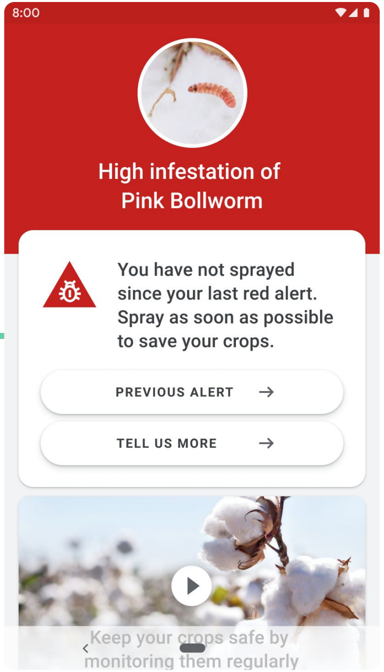


The pests are detected and counted by the AI from the image uploaded.



25

An advisory is then generated on the app.



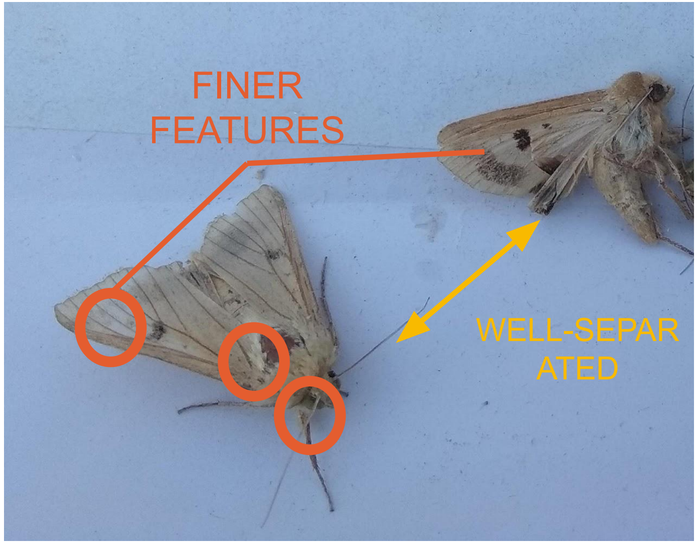
App Flow

Data Challenges

Data issues should be considered the **norm** when building applied AI solutions

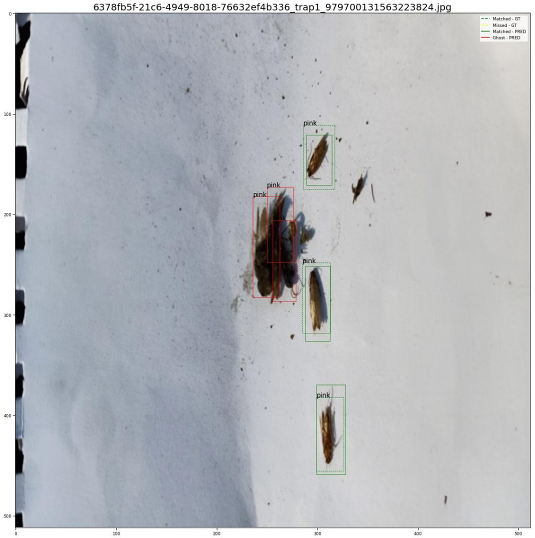
- Annotation problems: dead, trapped pests; very small; low-resolution images
- Data diversity due to varying agro-climactic zones and user behavior
- Seasonality and uncertainty in pest occurrence

What entomologists are trained to see



Trapped insect data

Occlusions



Neural Network Architecture

Multi-task learning with shared VGGNet backbone

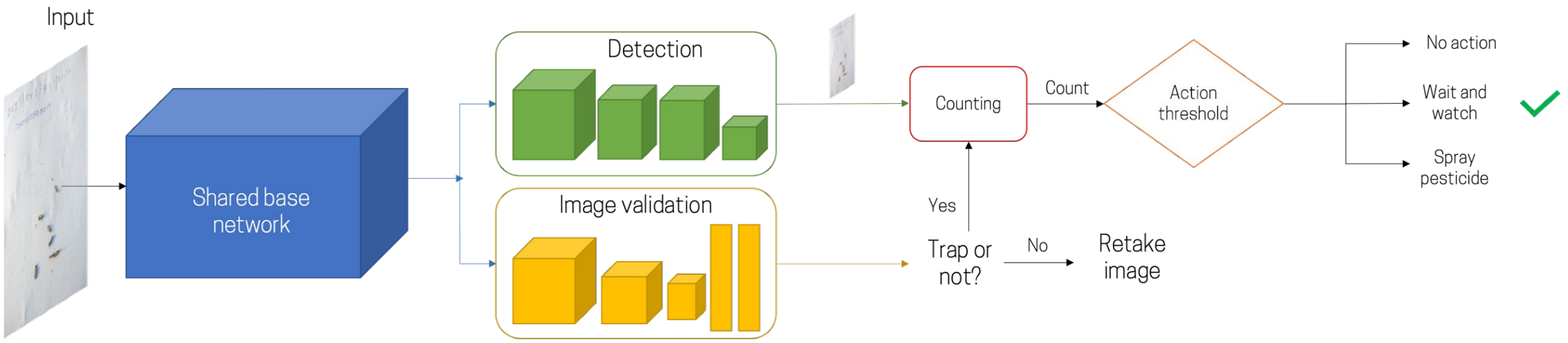


Image Validation: Ensure that the input is a valid trap image

Detection: Identify and count the pests in the image

Models for American bollworms (**ABW**), pink bollworms (**PBW**), Jassids

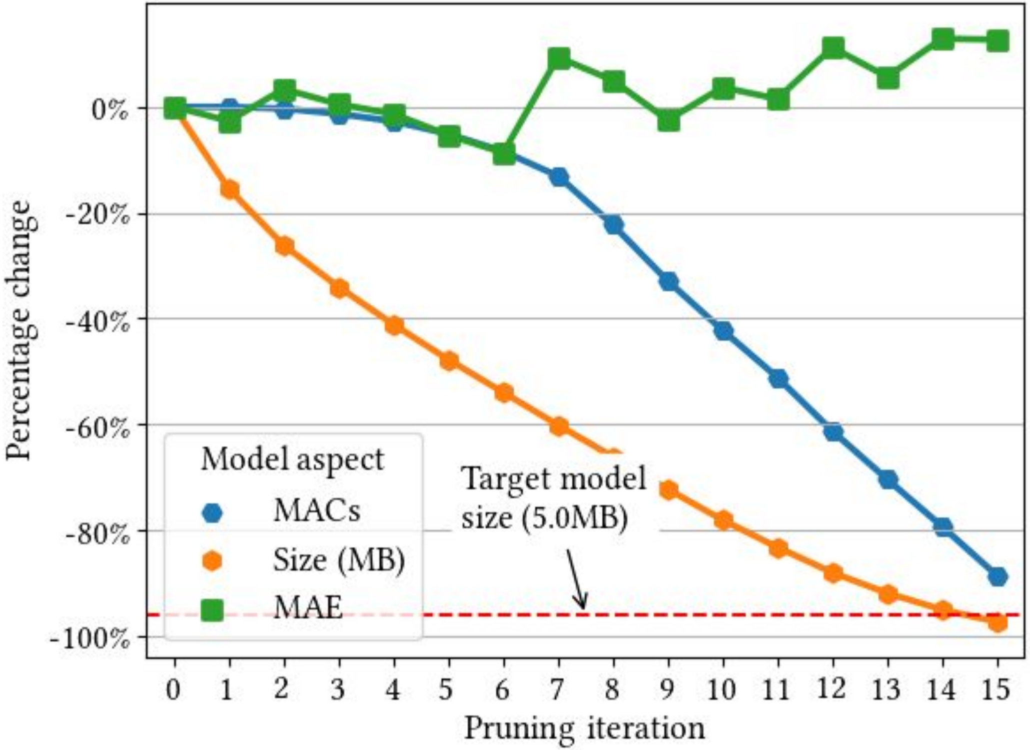
Results on Pest Count

Accuracy as a function of training set size

Pest	Images		AP_{50}	F_1	
	in class	total		per box	per image (SD)
PBW	77	246	0.753	0.771	0.814 (0.17)
	170	493	0.843	0.853	0.878 (0.13)
	320	986	0.888	0.888	0.907 (0.10)
	673	1972*	0.902	0.904	0.925 (0.09)
ABW	169	246	0.941	0.933	0.940 (0.11)
	323	493	0.970	0.955	0.957 (0.08)
	666	986	0.985	0.966	0.964 (0.07)
	1299	1972*	0.992	0.973	0.967 (0.08)

Compute in Low-Resource Settings

Model memory footprint vs Error tradeoff



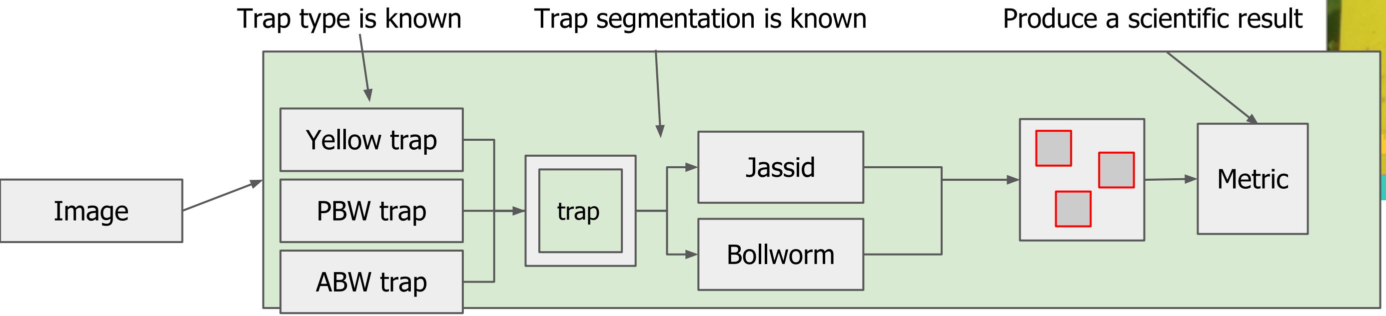
SMALL INCREASE IN MEAN ABSOLUTE ERROR FROM 0.91 TO 1.02

WE REDUCE OUR MODEL SIZE FROM 250MB TO 5MB

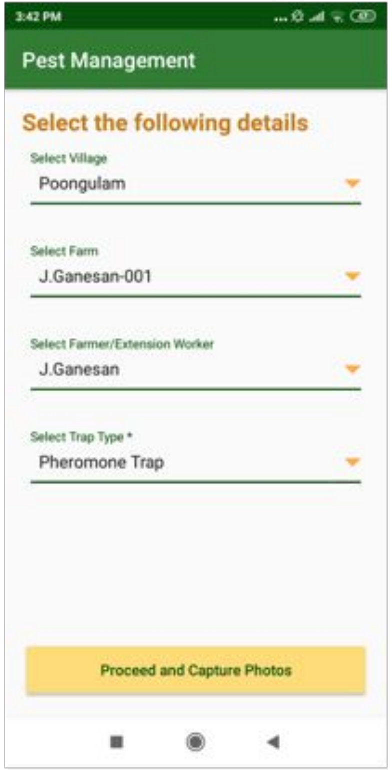
Deployment (Kutch, Wardha, Adilabad, Rangareddy)

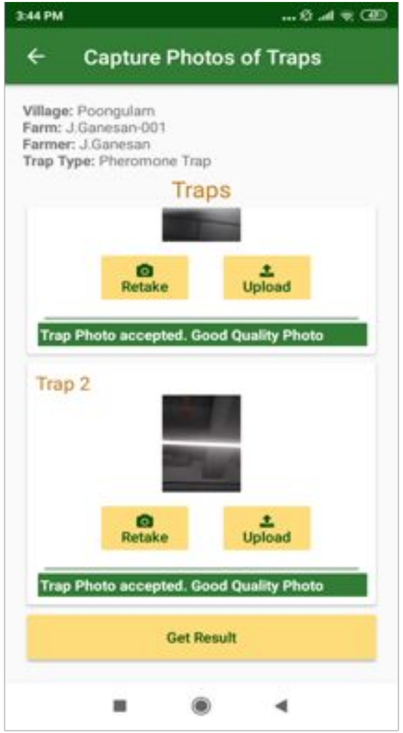
End to End solution is much more than the model

- Collect data
- Identify image quality
- Identify trap location
- Detect/count pests
- Provide advisory
- Integrate with extension program IT workflow



Pest Detection Pipeline in Practice





Solution Impact - Lead Farmers

Study conducted independently by E&Y

	Lead	Comment (if any)
Yield Per Acre	Inconclusive findings	On the yield for lead farmers, results were inconclusive due to low statistical power. This does not imply null impact on yield, and could be a comment on the number of observations. To understand the impact of the programme on yield therefore, a similar study at a later duration with a larger sample could be undertaken.
Pesticide Cost Per Acre	Saved INR 1,125 per acre** Accounts for 26% of their pesticide cost in 2019	NA
Selling Price Per Acre	Price gain per acre of INR 903** Accounts for 2.5% of their selling price in 2019	NA
Net Gain	Profit rise of INR 2,028 per acre Translates to ~20.5% increase in the 2019 profit margin	NA

Solution Impact - Cascade Farmers

Study conducted independently by E&Y

	Cascade	Comment (if any)
Yield Per Acre	Improved by 1.165 quintals/acre*** Accounts for 11% of yields in 2019	NA
Pesticide Cost Per Acre	Saved INR 1,737 per acre*** Accounts for 38% of pesticide cost in 2019	The relatively higher delta for cascade could be owing to catch-up
Selling Price Per Acre	Price gain per acre of INR 1050* Accounts for 2% of the selling price in 2019	The relatively higher delta for cascade could be owing to catch-up
Net Gain	Profit rise of INR 2,909 per acre Translates to ~26.5% increase in the 2019 profit margin	For profit calculation for cascade farmers, conservative specifications were taken. Since the result on selling price was significant at a higher level, we did not include it in our calculation for profit.

Next Steps

Ongoing and future work in AI-based pest management

- **Metadata collection**
- **Fixing annotation issues**
    - Expert inter-annotator agreement
    - Annotation protocol development
    - Scaled annotation through external agencies
- **End-to-end accuracy improvements**
    - Input validation
    - Object detection
- **Scaling of the solution**
- **“Trap-free” approaches**
    - Multiple signals: geo, weather, soil, satellite, ..
    - Multiple crops, multiple pest types
    - Large data collection for ground truthing

HEALTH CASE STUDY

Predicting Risk of Loss to Follow Up (LFU) in TB patients

Introduction

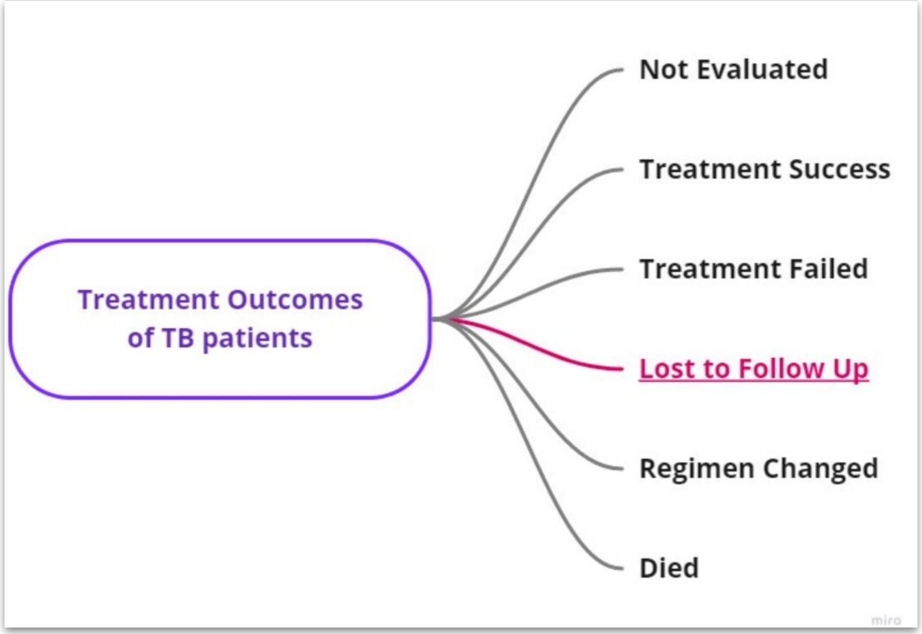
The **World Health Organization**, through the End Tuberculosis (TB) Strategy, aims to end the global TB epidemic, with targets to **reduce TB deaths by 95%** and to **cut new cases by 90% by 2035.**

With its ambitious **National Strategic Plan (NSP) 2017-2025**, the Government of India has planned to **achieve reduction in TB deaths by 90%** and to **reduce TB incidence by 80% by 2025**.

- The last four years have seen several policies and interventions augmenting the ambitious target of ending TB in India. However, TB treatment success rate in India hovers around **81.2%**, **well short of the target** set under NSP.

- India has the largest database of TB patients in the world - **Nikshay**, which has longitudinal data of every notified TB patient from diagnosis to treatment outcome.

What is Lost to Follow Up (LFU) ?



- Treatment Success of TB patients highly depends on the adherence to the treatment regimen.
- **Lost to follow up (LFU)** patient is defined as a TB patient whose treatment was interrupted for one or more consecutive month(s).

Early prediction of LFU and subsequent intervention is critical because LFU patients:

- Are silent transmitters of Tuberculosis
- Have higher risk of development and amplification of drug resistant TB
- Lead to increase in overall mortality
- Have increased risk of other morbidities

Our Solution to Predict and Reduce LFUs

WADHWANI AI

Patient Data from Nikshay
(Notification, Comorbidity, Contact Tracing, Lab and Adherence Data)

Contextual Data
(Population Density, Health Facility per district, Literacy, Socio economic, Access to health facilities)

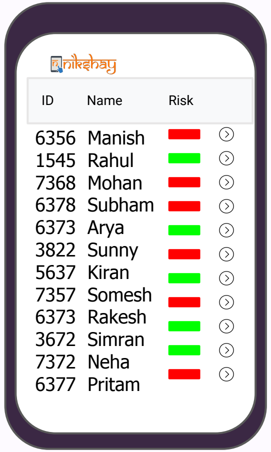
nikshay

Risk Prediction Using AI Model

At time of treatment initiation without adherence info

Risk Information and Proposed interventions
integrated with current system
(Visit, Call, SMS , etc)

Guide field health workers for differentiated care and intensified monitoring for high risk patients



Our approach uses an AI-based predictive model for early identification of patients at risk of LFU at the time of *treatment initiation*, in order to enable differentiated care

WADHWANI AI

- ➔ Use AI Model trained on patient and contextual data
- ➔ Generate risk score to predict high risk LFUs
- ➔ Propose programmatic interventions corresponding to risk level
- ➔ Monitor and analyse the efficacy

NIKSHAY

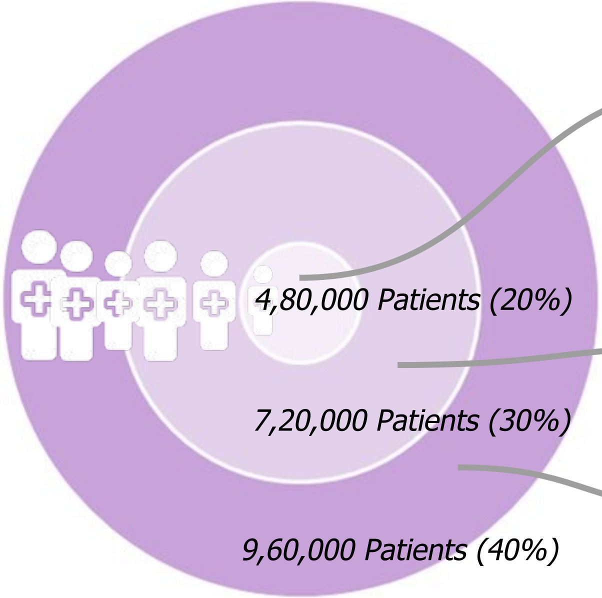
- ➔ Make risk stratification available to the end user to take up required intervention
- ➔ Provide ability to log activities in accordance with the new interventions

Solution Benefits

Process	Manual Intervention	Quality Care	Efficiency	Prediction
Provide proactive process that **identifies patients that are potential LFUs at an early stage** and enable timely intervention	**Reduce Manual Intervention** in identifying and prioritising High Risk patients	**Enable differentiated or stratified care** based on data driven measure and deliver high quality care to those who need it the most	**Improve intervention efficiency** with timely notification and targeted audience care	**Facilitate better future decisions** by tracking risk and intervention data

Potential Impact with Current Performance*

<u>Based on cases from 2019</u> : Total Patients : 24,00,000 | Total LFUs ~ 1,00,000 | Deaths : 8% | Drug Resistant : 37%



Targeting 20% : Sensitivity = 0.57

LFUs Identified	Controlled DRTB (37% of LFU identified)	Death Reduction (8% of LFU identified)
57,000	21,090	4,560

Targeting 30% : Sensitivity = 0.69

69,000	25,530	5,520

Targeting 40% : Sensitivity = 0.77

77,000	28,490	6,160

Sensitivity % : Refers to the % of total LFUs correctly classified as LFU by the model
*Note : The potential numbers are subjected to high TB program efficiency on ground

Interventions Overview

Find more information on interventions and metrics

WHEN

- At treatment initiation
- Bi-weekly
- Weekly
- Monthly
- As needed

WHERE

- PHIs (PHCs, Health Posts, etc)
- Community
- Home
- Panchayat Raj Institutions
- Local NGOs

Patient

WHO

- STO and DTO
- STS and TBHV
- Medical Officer
- Staff nurse/ Pharmacist
- Treatment supporter / MPW
- Community leaders

WHAT

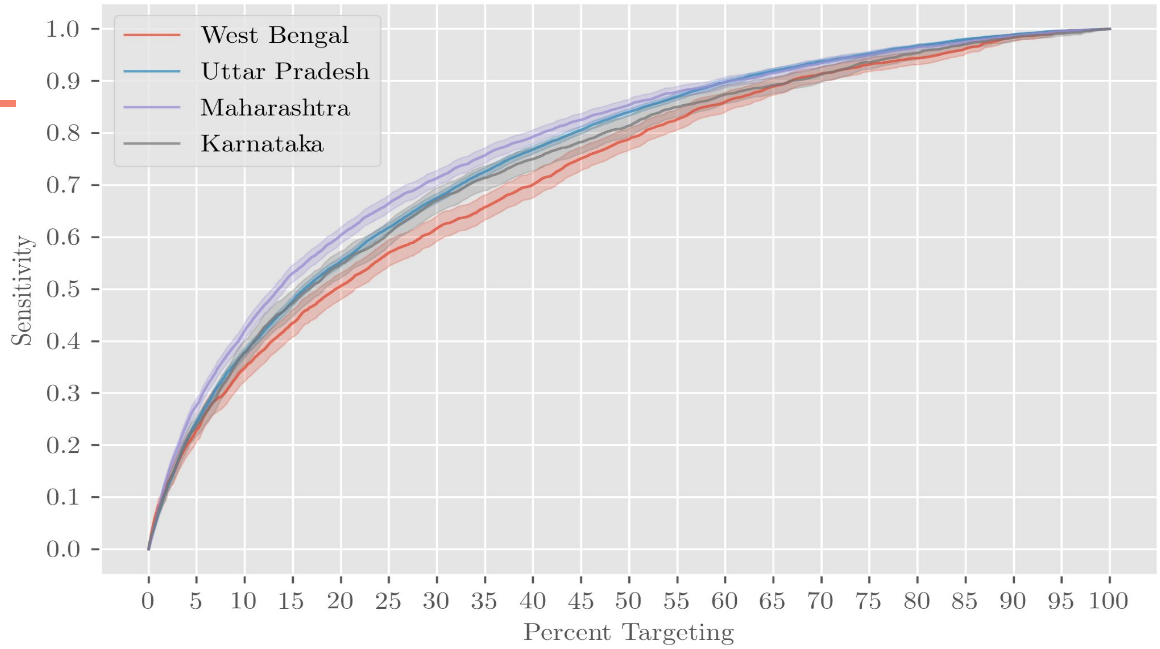
- Personalized Counselling
- Mandatory treatment supporter and changing custody of medication
- Follow up visits
- Linkages
- Supportive supervision based on Digital adherence tools (DAT)

These interventions are based on risk score predicted by the AI Model. We will work with CTD to ensure success in actual settings

State-wise Performance

- Model performance strongest in Maharashtra
- Significant lift over rules-based model across states in number of additional LFUs identified with the same % targeted

Check detailed information on rule based models



OVERALL	Total patients	Total LFUs	Sensitivity @ 20%	Baseline@20% (Rules-based)	Lift over rules-based
Maharashtra	91,041	3,567	0.603 +/- 0.016	0.406	**49%**
Uttar Pradesh	233,046	11,069	0.553 +/- 0.009	0.300	**84%**
Karnataka	41,933	1,436	0.548 +/- 0.026	0.384	**43%**
West Bengal	49,325	1,353	0.508 +/- 0.026	0.319	**59%**

Overall Results: Model Robustness

WADHWANI AI

	Initial POC (Feb’21)	Passive Evaluation (Jun’21)*
Total Patients	38,243	9,43,654
Test Patients	3,800	4,31,932
Sensitivity @20	58%	57%
Sensitivity @30	65%	69%
Sensitivity @40	75%	77%
PPV @20	13%	12%
PPV @30	10%	10%
PPV @40	9%	8%

➔ **Sensitivity@k :** Refers to the **% of total LFUs correctly classified as LFU by the model** when the top k% patients are targeted

➔ **PPV(Positive Predictive Value)@k :** Refers to **% of real LFUs among the patients targeted** when the top k% patients are targeted.

***Passive evaluation consists of all patients from 2019 and some from 2020. For detailed results , click here**

Impact of Data Registers and Data Availability on Performance

Based on combination of Nikshay registers

Register Combination	Sensitivity	Increase in performance (%)
With notification register (Base)	0.526	0
With notification and comorbidity registers	0.54	2.66
With notification, comorbidity, and contact tracing registers	0.539	2.47
With notification, comorbidity, contact tracing, and patient lab registers	0.527	-0.19

Based on data delay from treatment initiation

Delay from the start of treatment initiation	Sensitivity	Increase in performance (%)
0 Day	0.5	0
7 Days	0.486	-2.8
21 Days	0.555	11

Assumption : The results are based on Jan-June 2019 nikshay patient data and will be revised once we incorporate 2020 nikshay data into the model.

Key Risks and Challenges

Risks	Causes	Mitigation
Delayed data entry in Nikshay	HWs may be busy with other work or there are delays in patient follow ups. Delay in data entry can affect the model output score and ultimately the LFU Risk Status	1. Allow 1 week to fill up notification covariates and 4 weeks to complete data entry from the start of treatment initiation. 2. Send alerts to workers and higher officials, if not filled up. 3. Use PATH resources in ensuring on-time data collection
TUs with too many or too few high-risk patients	As the district thresholds used for risk stratification are historical, there may be TUs with too many or too few high-risk patients	Educate and enforce a process for threshold changing requests to Nikshay Team
Data quality/reporting issues	With numerous challenges being faced for data reporting and entry, we might expect data quality issues like incompleteness or irrelevant values in the raw data	1. Conduct a literature survey on the accuracy of reported TB treatment outcomes in India and make the appropriate adjustments in the algorithm 2. Explore conducting a ground-truthing study in select geographies
Target Performance not achieved	Low predictive power at the initiation stage, without info on adherence	1. Expand the scope of the solution to include follow-up information and longitudinal data on treatment adherence
Low Effectiveness of interventions	Effectiveness of interventions for high-risk groups such as migrants unclear	1. Understand delay in data entry by geography 2. Monitor and log activities for interventions 3. Notify users

Paths to success with applied AI

- **The devil is in the data!** Collect well, annotate well, sample well, understand the data.
  - When annotations / labels are not trustworthy, use multiple annotations, expert annotations.
  - Aim for complete, accurate features as far as possible. Compromise: large datasets.
  - Use metadata to advantage.
  - Use common sense! What features are likely to be useful for prediction? Are they feasible to collect? If not, are proxies feasible?

- **Build models that are robust**, and withstand the test of time.
  - Evaluate across cohorts, across time, keep evaluating and retraining until results stabilize.
  - Favour simple, interpretable models over large models (if performance is similar).

- Understand assumptions and limitations of the deployment environment BEFORE starting work on AI models. **Think backwards from deployment.**

Thank you!

WADHWANI AI

www.wadhwaniai.org