

# Copyright Notice

These slides are distributed under the Creative Commons License.

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

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

# Copyright Notice

These slides are distributed under the Creative Commons License.

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

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

# Copyright Notice

These slides are distributed under the Creative Commons License.

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

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



DeepLearning.AI

# C1W2 Slides

---

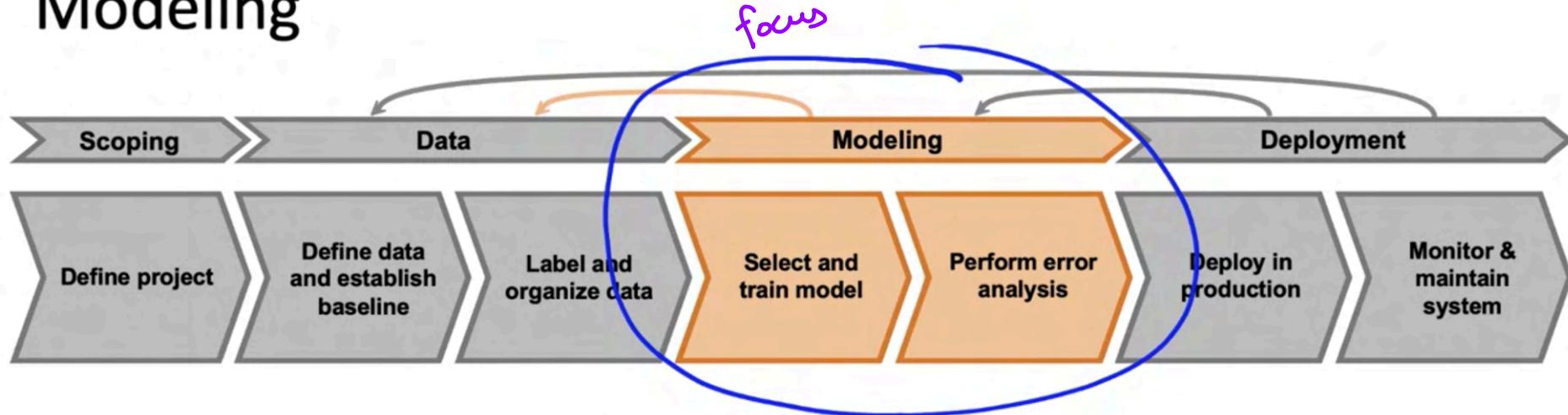


# Select and train model

---

## Modeling overview

# Modeling



Model-centric AI  
development

Data-centric AI  
development



# Select and train model

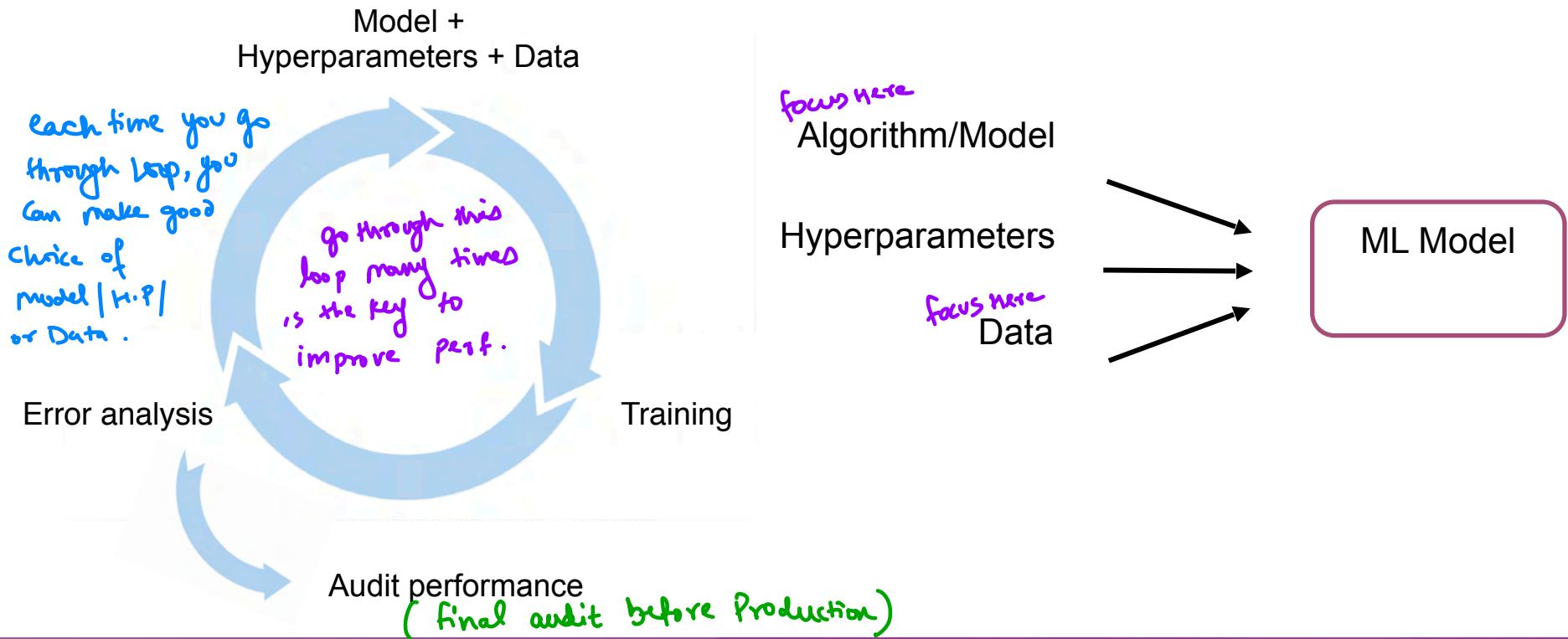
---

DeepLearning.AI

## Key challenges

**AI system = Code + Data**  
(algorithm/model)

# Model development is an iterative process



# Challenges in model development

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

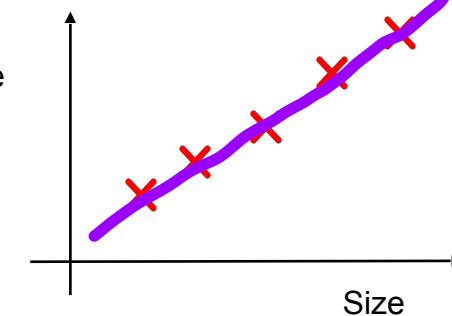


2. Doing well on dev/test sets. *(earlier People used to focus here)*



3. Doing well on business metrics/project goals.

*(But now focus is here)*





DeepLearning.AI

# Select and train model

---

## Why low average test error isn't good enough

What's more than that?

# Performance on disproportionately important examples



## Web Search example

↳ try giving higher weight to imp. examples.  
(Here navigational query)

"Apple pie recipe"

"Latest movies"

"Wireless data plan"

"Diwali festival"

"Stanford"

"Reddit"

"Youtube"



### Informational and Transactional queries

(No exact web page to be fetched at top.  
→ Can be ranked.)

### Navigational queries (clear navigation)

→ Don't return anything else here.  
→ Don't mess up with Navigation queries  
(Not acceptable) else loses users trust.

# 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. [Not acceptable the discrimination]

## Example: Product recommendations from retailers

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

- ~~Unacceptable~~
- irrelevant recommendation to specific ethnicity.
  - only large brands are recomm. (ignored small brands)

# Rare classes

Skewed data distribution

99% negative

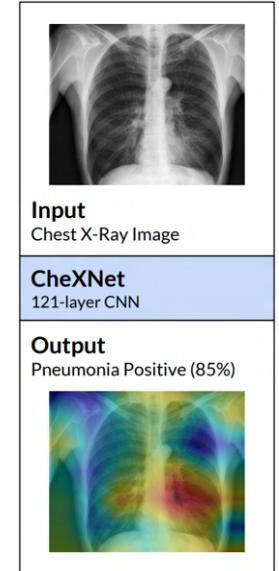
1% positive  
(disease)

↓  
Model  
print("0") ←

Accuracy in rare classes

Condition	Performance
Effusion	0.901 ←
Edema	0.924
Mass	0.909
Hernia (Rare)	0.851 ←

Model could ignore Hernia cases since in test set only v. few Hernia images are present so even if Model misclassify them , it won't affect Majorly .



→ we need to go beyond test set.

# 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!"



DeepLearning.AI

# Select and train model

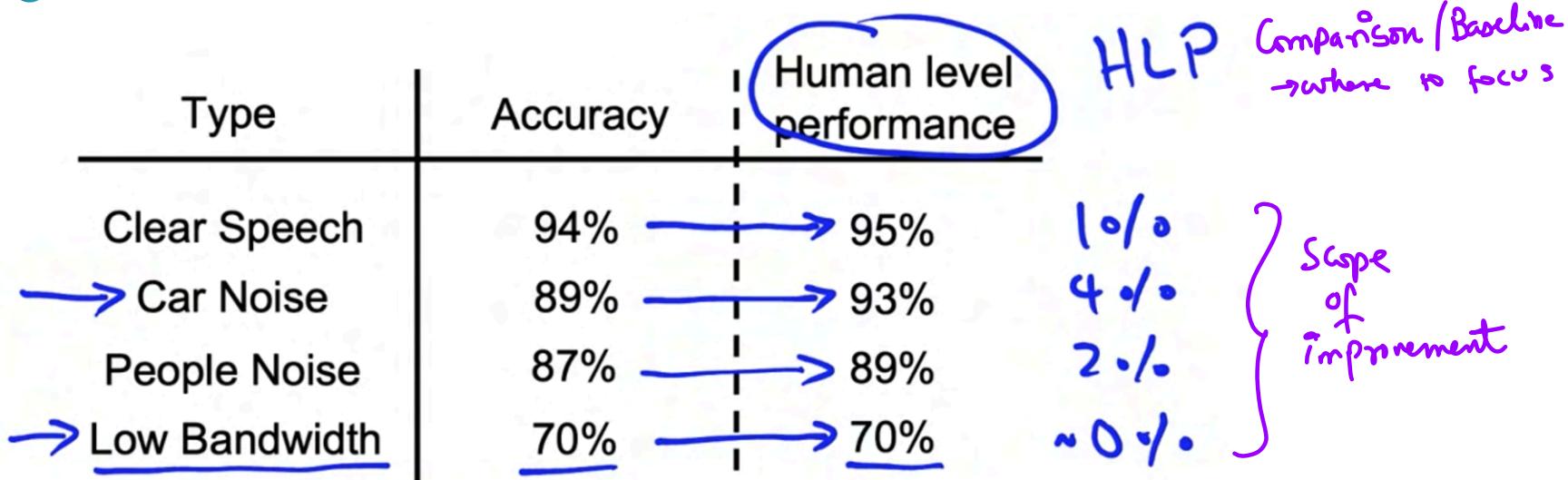
---

Establish a  
baseline *Performance*

# Establishing a baseline level of performance



Speech recognition example:



# Structured and unstructured data

Unstructured data  
*(Humans are good)*

Image



Audio



Text

This restaurant was great!

Here, Baseline can be

HELP

Structured Data

User Id	Purchase	Number	Price
3421	Blue shirt	5	\$20
612	Brown shoes	1	\$35

Price	Product
3421	Red skirt

Here HELP is less common for baseline.

# Ways to establish a baseline

- Human level performance (HLP)
- Literature search for state-of-the-art/open source (*other's perf.*)
- Older system

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

# Select and train model

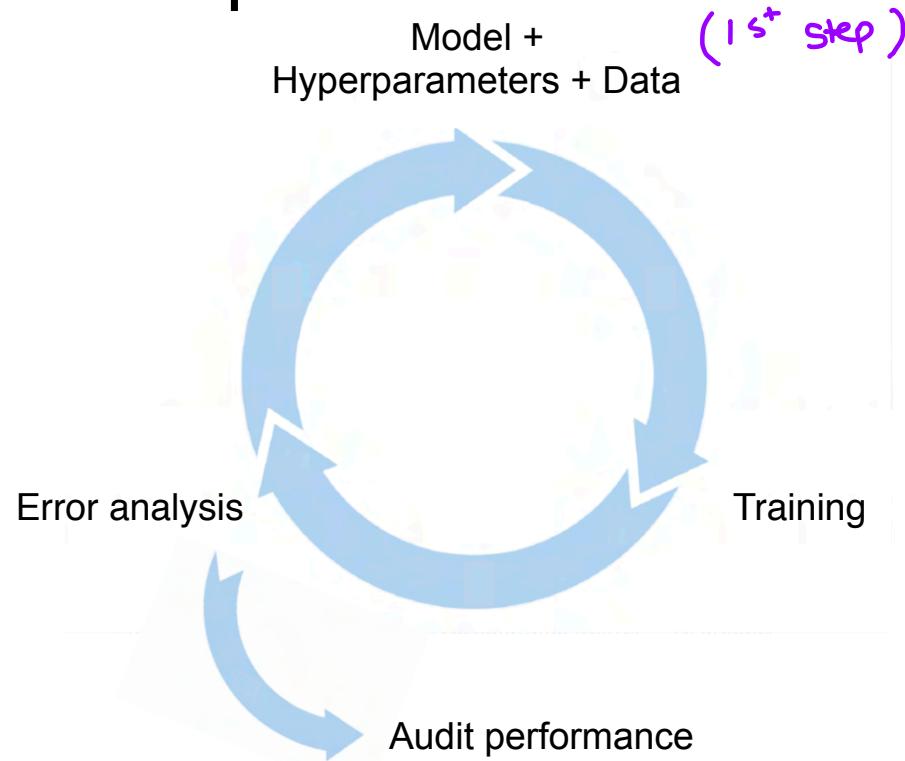
---



DeepLearning.AI

## Tips for getting started

# ML is an iterative process



# Getting started on modeling

- Literature search to see what's possible. (*Courses, blogs, open-source Projects*)  
→ get started quickly.
- Find open-source implementations if available.
- A reasonable algorithm with good data will often outperform a great algorithm with not so good data.

# Deployment constraints when picking a model

Should you take into account deployment constraints when picking a model?

**Yes**, if baseline is already established and goal is to build and deploy.

**No**, if purpose is to establish a baseline and determine what is possible and might be worth pursuing. *(Not necessarily)*

# Sanity-check for code and algorithm

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

- Example #1: Speech recognition

*1 audio sample only*

audio transcript  
X → Y



- Example #2: Image segmentation

*train 1 image to see if it's overfit or not*



- Example #3: Image classification

*train → 10, 100 images →*

# Error analysis and performance auditing

---



DeepLearning.AI

## Error analysis example

→ Where should you invest your time improving perf.

# Speech recognition example

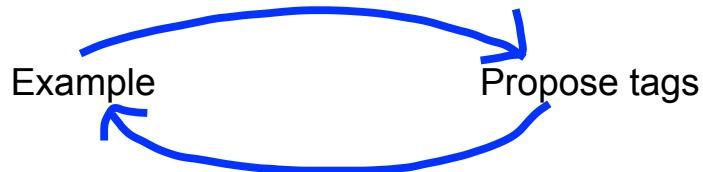
Manual check 100 mislabelled e.g from dev set

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"	✓	✓	✓

Type of data in training set  
additional tag

Code analysis is manual, But some MLops tools exists which make this process easier

# Iterative process of error analysis



## Visual inspection:

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



## Product recommendations:

- in irrelevant recommendation case these  
are user demographics (Young women or old men)  
or Product features /Category

Then you'll be able to improve model performance .

# Useful metrics for each tag

- What fraction of errors has that tag? 12%. audio → car tag (how imp this to solve problem)
- Of all data with that tag, what fraction is misclassified? e.g. 18%.
- What fraction of all the data has that tag?
- How much room of improvement is there in that tag?

# Error analysis and performance auditing

---



DeepLearning.AI

Prioritizing what to work on

# Prioritizing what to work on

Type	Accuracy	Human level performance	Gap to HLP	% of data	Overall Inc <sup>↑</sup>
Clean Speech	94%	95%	1%	60%	→ 0.6%
Car Noise	89%	93%	4%	4%	→ 0.16%
People Noise	87%	89%	2%	30%	→ 0.6%
Low Bandwidth	70%	70%	0%	6%	→ ~0%

# 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. (imp.)
- How easy is to improve accuracy in that category. for Car Noise like - augmentation
- ★ • How important it is to improve in that category. Car noise is imp.

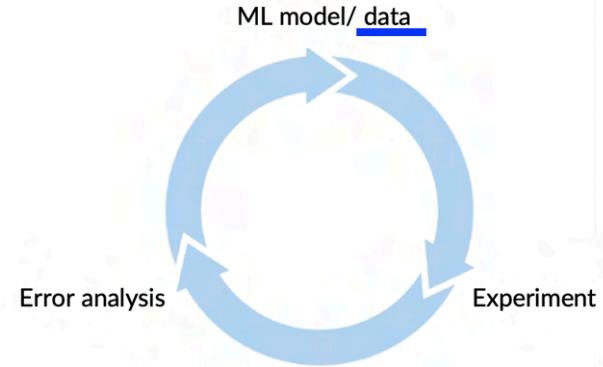
# Adding data

For categories you want to prioritize: (for car noise)

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

→ Data collection can be expensive & time consuming.  
→ With this approach you're focused on where to spend time.

Type	Accuracy	Human level performance	Gap to HLP	% of data
Clean Speech	94%	95%	1%	60%
→ Car Noise	84%	93%	4%	40%
→ People Noise	87%	84%	2%	30%
Low Bandwidth	70%	70%	0%	6%



# Error analysis and performance auditing

---



DeepLearning.AI

Skewed  
datasets

imbalanced  
data

# Examples of skewed datasets

[Accuracy is Not imp.]

(Mobiles)

## Manufacturing example



99.7% no defect

$$y=0$$

0.3% defect

$$y=1$$

print("0")  
99.7%



Medical Diagnosis example: 98% of patients don't have a disease



Speech Recognition example: In wake word detection, 96.7% of the time wake word doesn't occur

# Confusion matrix: precision and recall

		Actual	
		$y=0$	$y=1$
Predicted	$y=0$	905 TN	18 FN
	$y=1$	9 FP	68 TP

$\hookrightarrow 914$        $\hookrightarrow 86$

$TN$ : True Negative

$TP$ : True Positive

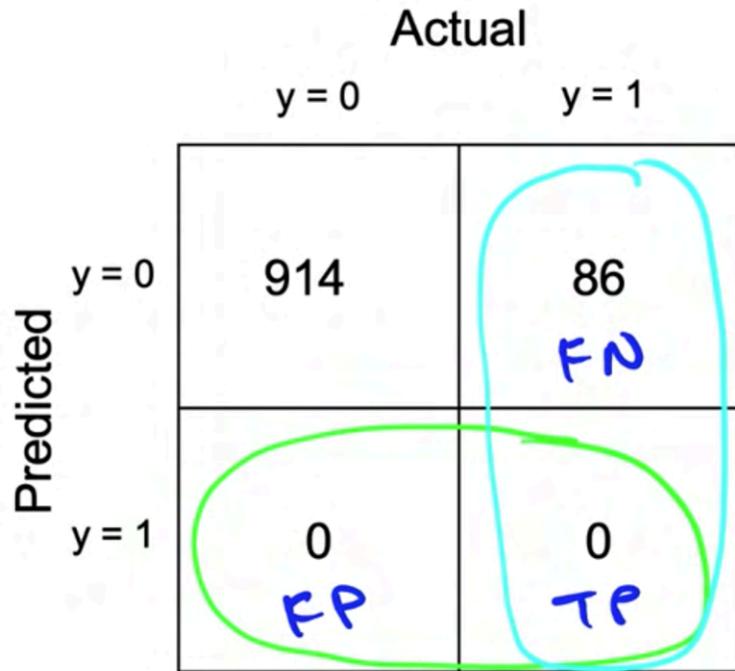
$FN$ : False Negative

$FP$ : False Positive

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{68}{68+9} = 88.3\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{68}{68+18} = 79.1\%$$

# What happens with print("0")?



$$\text{Precision} = \frac{TP}{TP + FP} = \frac{0}{0+0}$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{0}{0+86} = 0\%$$

# Combining precision and recall – $F_1$ score

	Precision ( $P$ )	Recall ( $R$ )	$F_1$
Model 1	88.3	79.1	83.4 %
Model 2	97.0	7.3	13.6 %

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

# Multi-class metrics

using  $F_1$ , you can decide which type of defect to work on!

Classes: Scratch, Dent, Pit mark, Discoloration

Defect Type	Precision	Recall	$F_1$
Scratch	82.1%	99.2%	89.8%
Dent	92.1%	99.5%	95.7%
Pit mark	85.3%	98.7%	91.5%
Discoloration	72.1%	97%	82.7%

Macro - weighted?

Micro - weighted?

# Error analysis and performance auditing

---



DeepLearning.AI

## Performance auditing

# Auditing framework

Check for accuracy, fairness and bias.

1. Brainstorm the ways the system might go wrong.
  - Performance on subsets of data (e.g., ethnicity, gender).
  - Prevalence of specific errors/outputs (e.g., FP, FN).
  - Performance on rare classes.
2. Establish metrics to assess performance against these issues on appropriate slices of data. (*Subset data - specific ethnicity, gender or Scratchup phones*)
3. Get business/product owner buy-in. “TFmp”

# Speech recognition example

1. Brainstorm the ways the system might go wrong. (*problem dependent*)
  - Accuracy on different genders and ethnicities.
  - Accuracy on different devices.
  - Prevalence of rude mistranscriptions.
2. Establish metrics to assess performance against these issues on appropriate slices of data.
  - Mean accuracy for different genders and major accents.
  - Mean accuracy on different devices.
  - Check for prevalence of offensive words in the output.



DeepLearning.AI

# Data iteration

---

## Data-centric AI development

if error analysis suggests to work on improving performance  
for specific issue. → try improving data quality.

# Data-centric AI development

## Model-centric view

Collect what data you can, and develop a model good enough to deal with the noise in the data.

Hold the data fixed and iteratively improve the code/model.

## Data-centric view

Useful Many app.

The consistency of the data is paramount. Use tools to improve the data quality; this will allow multiple models to do well.

*Hold the code fixed and iteratively improve the data.*

# Data iteration

---



DeepLearning.AI

A useful picture of data  
augmentation

# Speech recognition example

Different types of speech input:

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

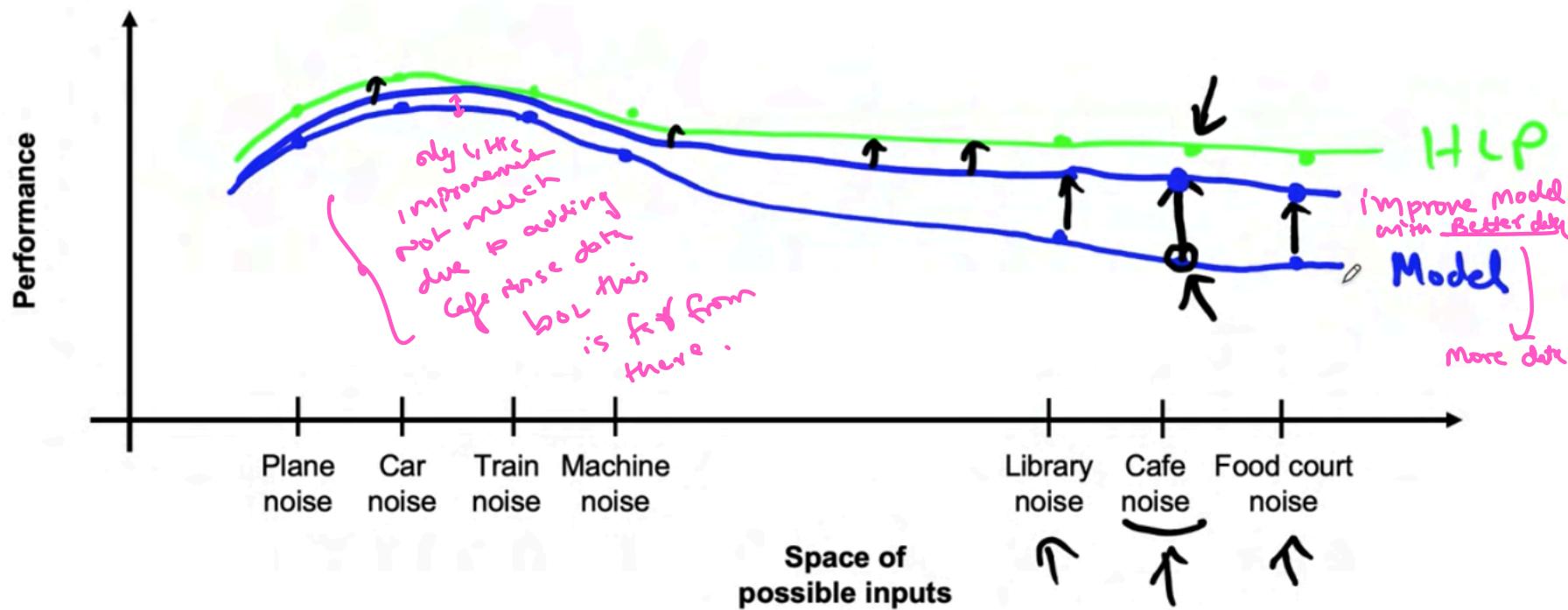


Similar - mechanical noise



Similar - people talking

# Speech recognition example





DeepLearning.AI

# Data iteration

---

## Data augmentation

*Specially for unstructured data*

# Data augmentation

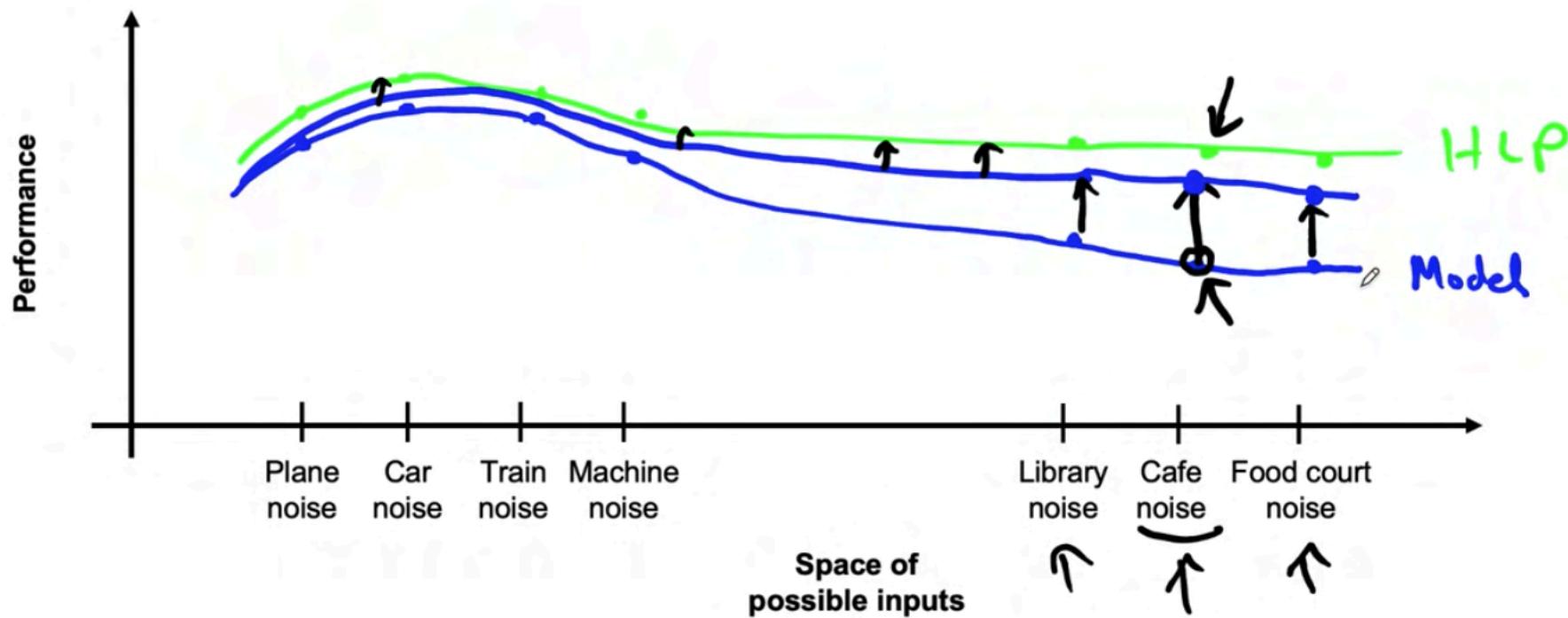
Goal:

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

Checklist:

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

# The rubber sheet analogy



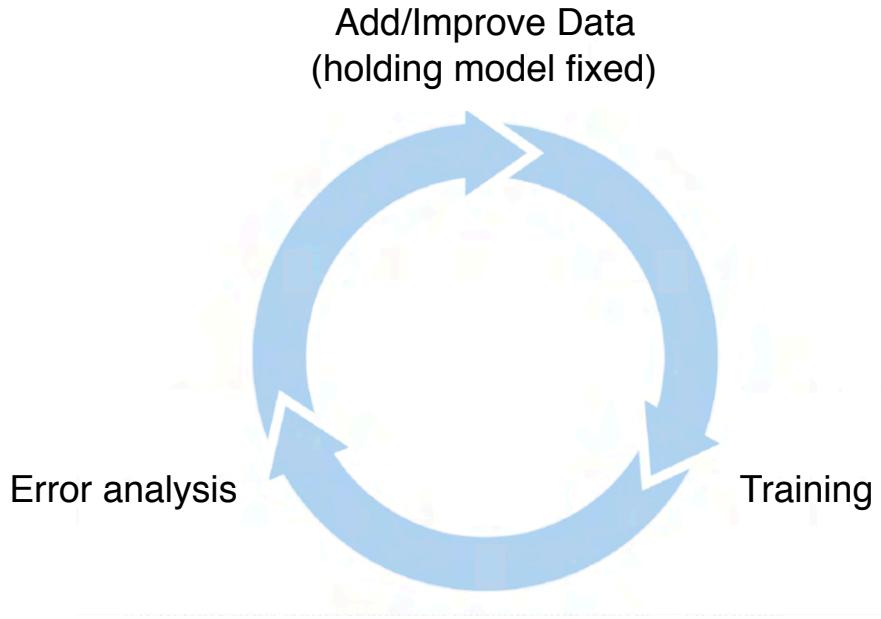
# Image example



→ GANs Can be overkill

Artificially draw  
Scratch → Photoshop

# Data iteration loop





DeepLearning.AI

# Data iteration

---

Can adding  
data hurt?

usually for unstructured data  
→ No. But Greats

Adding data to  
training set  
might lead to  
diff dist  
in train | dev set.

# Can adding data hurt performance?

For unstructured data problems, if:

- The model is large (low bias).
- The mapping  $X \rightarrow Y$  is clear (e.g., humans can make accurate predictions).  
*(rare problem) I vs. 1*

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

originally 20% Cafe Noise

now 50% Cafe Noise

But if Model is large, it's acceptable & Model will do fine on this type of audio as well as other.

If Model is small then it will spend too much on Cafe noise & do Poor on other type of data.

# Photo OCR counterexample



1

high accuracy



I

low accuracy



?

ambiguous  
1? I?

Adding a lot of new "I"'s may skew the dataset and hurt performance



DeepLearning.AI

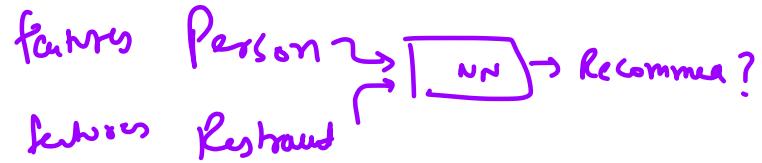
# Data iteration

---

## Adding features

for structure data  
→ creating new e.g.  
are difficult  
So add features

# Structured data



## Restaurant recommendation example

Vegetarians are frequently recommended restaurants with only meat options.  
( Popular | Hot | New )

Possible features to add?

- Is person vegetarian (based on past orders)? ( %. of order non-veg )
- Does restaurant have vegetarian options (based on menu)?

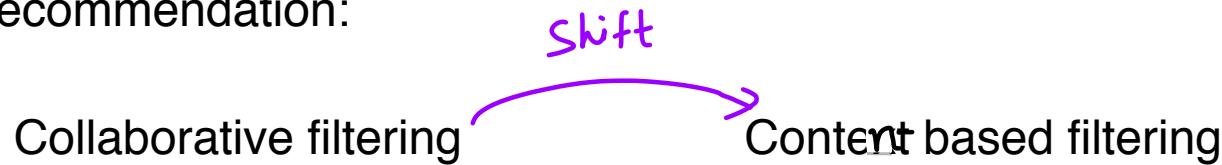
Person | Restaurant → "fixed" (add features to improve perf)

# Other food delivery examples

- Only tea/coffee
- Only pizza

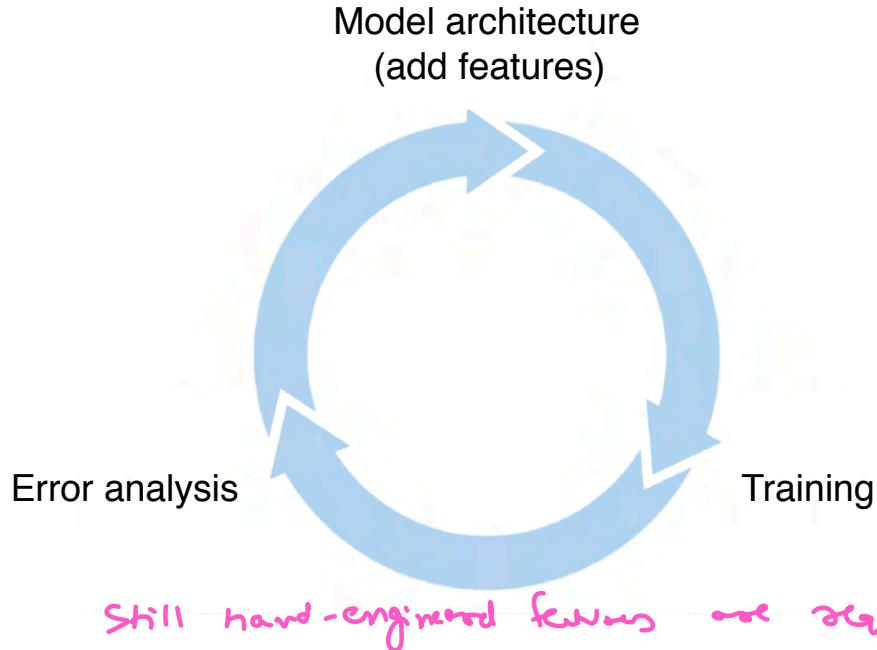
What are the added signals (features) that can help make a decision?

Product recommendation:



# Data iteration

h.n.p is hard on structured data.



- Error analysis can be harder if there is no good baseline (such as HLP) to compare to.
- Error analysis, user feedback and benchmarking to competitors can all provide inspiration for features to add.

# Data iteration



DeepLearning.AI

*Robust*

## Experiment tracking

# Experiment tracking

	Copy of trained model	
What to track?	Algorithm/code versioning Dataset used Hyperparameters Results	Tracking tools
Desirable features	Data needed to replicate results In-depth analysis of experiment results – Summary metrics Perhaps also: Resource monitoring, visualization, model error analysis	Text files Spreadsheet Experiment tracking system (MLflow)  Sagemaker Studio, Comet Weights & Biases

# Data iteration

---



DeepLearning.AI

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