

Week 2: Select and Train a Model

This week is about model strategies and key challenges in model development. It covers error analysis and strategies to work with different data types. It also addresses how to cope with class imbalance and highly skewed data sets.

[Focus: Modeling Stage] --> improve data in the most efficient way

1. Selecting and Training a Model

- **Key Challenges**

AI system = Code (algorithm/model) + Data

- 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. (may not sufficient) --> machine learning team
 3. Doing well on **business metrics/project goals**. --> business team

- **Why low average test error isn't good enough**

Just doing well on the test set is not enough for many production applications! --> try to build a machine learning system that solves the actual business or application needs

- Performance on disproportionately important examples
- 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.
- Rare Classes
 - Skewed data distribution
 - Accuracy in rare classes may have worse performance

- **Establish a Baseline**





Speech recognition example:

Type	Accuracy	Human level performance	HLP
Clear Speech	94%	95%	10/0
→ Car Noise	89%	93%	4/0
People Noise	87%	89%	2/0
→ <u>Low Bandwidth</u>	<u>70%</u>	<u>70%</u>	~0/0

--> Without human level performance, maybe it's shown that "Low Bandwidth" performance should be improved; however, with human level performance, "Car Noise" should be focused on, "Low Bandwidth" may not have much room to improve

◦ Unstructured and Structured Data

Unstructured data		Structured Data							
Image		User Id	Purchase	Number	Price				
Audio		3421	Blue shirt	5	\$20				
Text	<div>This restaurant was great!</div>	612	Brown shoes	1	\$35				
		<table><tr><th>Price</th><th>Product</th></tr><tr><td>3421</td><td>Red skirt</td></tr></table>				Price	Product	3421	Red skirt
Price	Product								
3421	Red skirt								

- Human are good at unstructured data tasks --> measuring HLP is a good way to establish a baseline
- Human are not good at looking at structured data to make predictions --> HLP is less useful

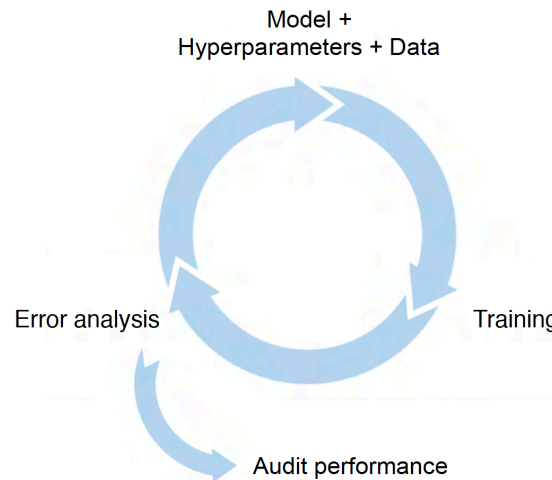
◦ Ways to establish a baseline

--> Baseline gives an estimate of the **irreducible error / Bayes error** and **indicates what might be possible**. --> efficient for prioritizing what to work on

- Human level performance (HLP) **[for unstructured data]**
- Literature search for state-of-the-art/open source --> use reports for a reference + start-point implementation

- Performance of older system

- **Tips for Getting Started**



- Getting started on modeling:
 - Literature search to see what's possible (courses, blogs, open-source projects)
--> for industry, read blog posts and pick something reasonable to get started quickly
 - Find open-source implementations if available. --> can establish a baseline efficiently
 - **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 (like computing constraints)?
 - **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.
- Sanity-check for code and algorithm **[Important before running the model on all the data]**
--> Try to **overfit a small training dataset** before training on a large one. (feed one image/one word/several examples) --> If a problem occurs, debug the code/algorithm/hyperparameters to make it pass this sanity-check test first, before moving to larger datasets.

2. Error Analysis and Performance Auditing

- **Error Analysis 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"	✓	✓	✓

--> help engineer understand whether the categories may be the source of more errors and be worthy of further effort and attention

- **[Typically done via manual process but there are emerging MLOps tools to help]**
- 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:

- User demographics
- Product features

- 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? --> measure HLP

• **Prioritizing What to Work On**

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

--> because 'Clean Speech' and 'People Noise' account for a large portion of data, it's worthwhile to work

- 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.
- After deciding on the category, to improve the average performance --> Add/Improve data for specific categories
 - Collect more data (or improve label accuracy)
 - Use **data augmentation** to get more data
 - Improve label accuracy/data quality

• Skewed Datasets

- Confusion matrix: precision and recall

		Actual	
		$y=0$	$y=1$
Predicted	$y=0$	905 TN	18 FN
	$y=1$	9 FP	68 TP
		↳ 914	↳ 86

- *TN: True Negative* --> actually negative and the algorithm predicts to be negative
- *TP: True Positive* --> actually positive and the algorithm predicts to be positive

FN: False Negative --> actually positive but the algorithm predicts to be negative

FP: False Positive --> actually negative but the algorithm predicts to be positive

[True/False: whether the algorithm predicts correctly; Positive/Negative: the algorithm prediction]

■ $Precision = \frac{TP}{TP+FP}$

$Recall = \frac{TP}{TP+FN}$ --> of all the actually positive examples, what fraction does the algorithm get right? --> if all predictions are '0', recall = 1

$F_1 Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$ --> combine precision and recall that emphasises which is worse
[harmonic mean between precision and recall] --> use F1 as a supplement of accuracy if the class is rare

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

--> Model 1 appears to be superior than Model 2

○ Multi-class Metrics

Classes: Scratch, Dent, Pit mark, Discoloration (maybe for phone factory)

[Many factories emphasize high recall]

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%

● **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**.
(Eg. Tensorflow has a package for tensorflow model analysis)
 3. Get business/product owner buy-in.
- Example

Speech recognition example

1. Brainstorm the ways the system might go wrong.
 - 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.