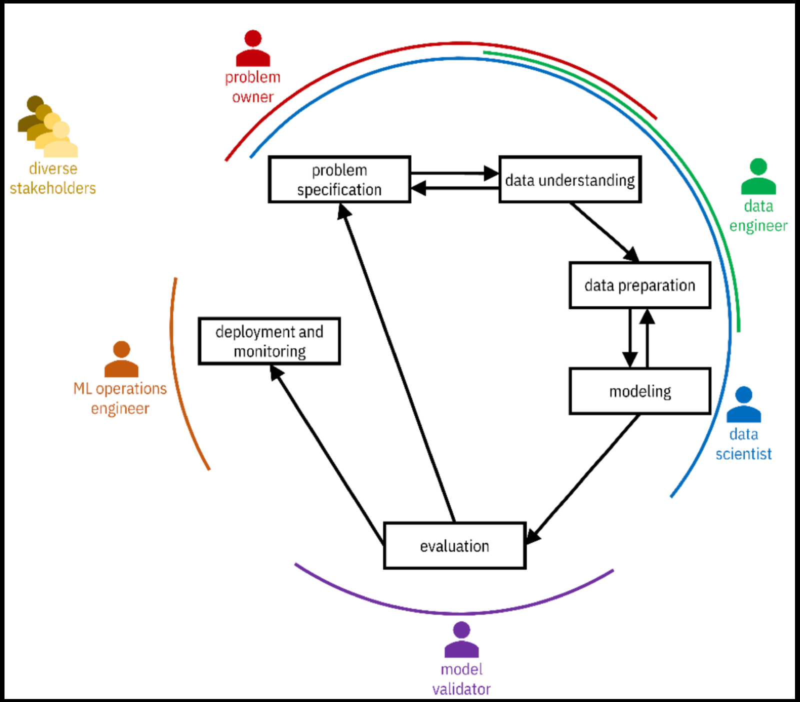
Machine Learning for Design

Lecture 5 - Part *a*

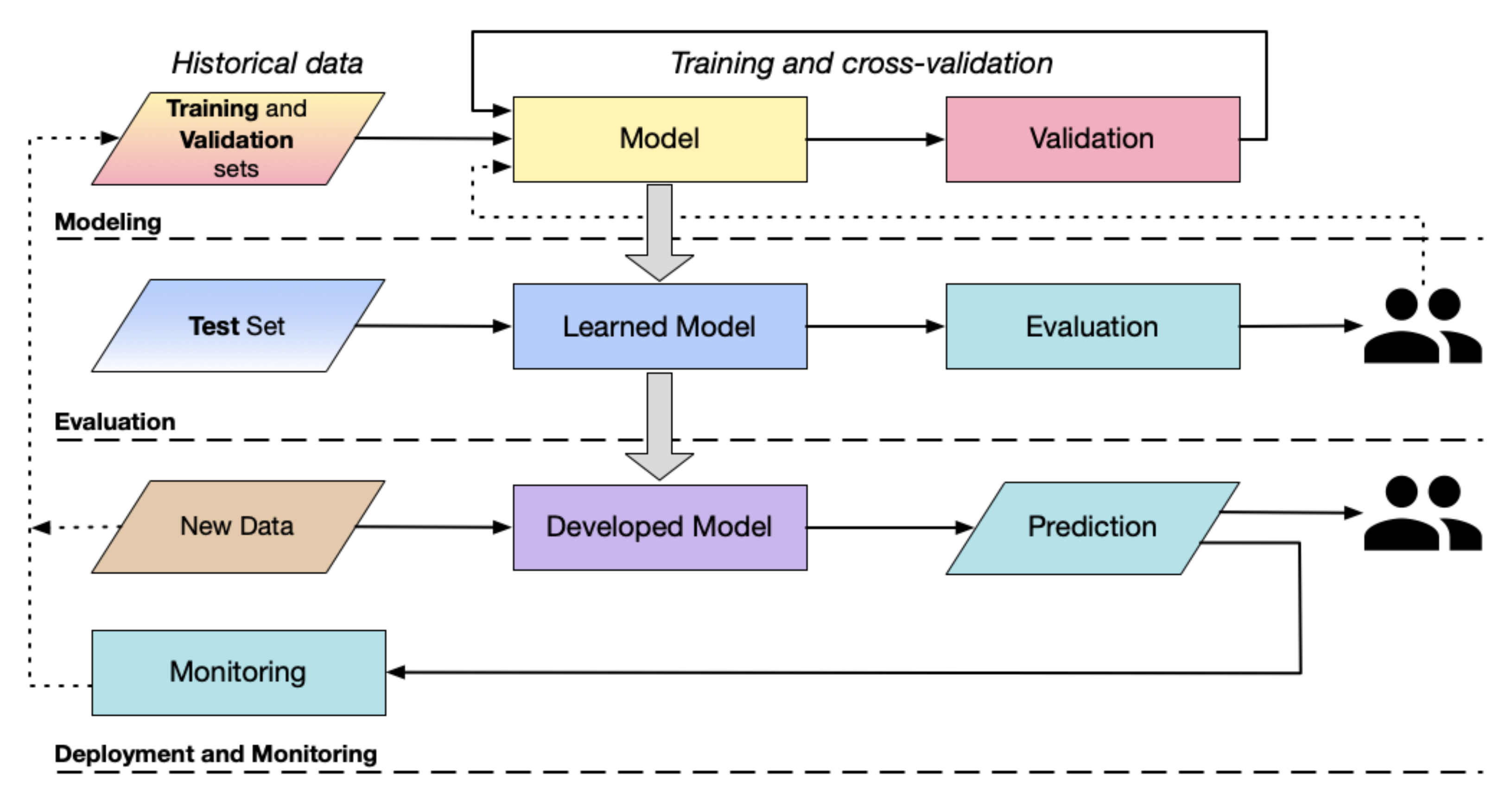
Training and Evaluation

Previously on ML4D

CRISP-DM Methodology

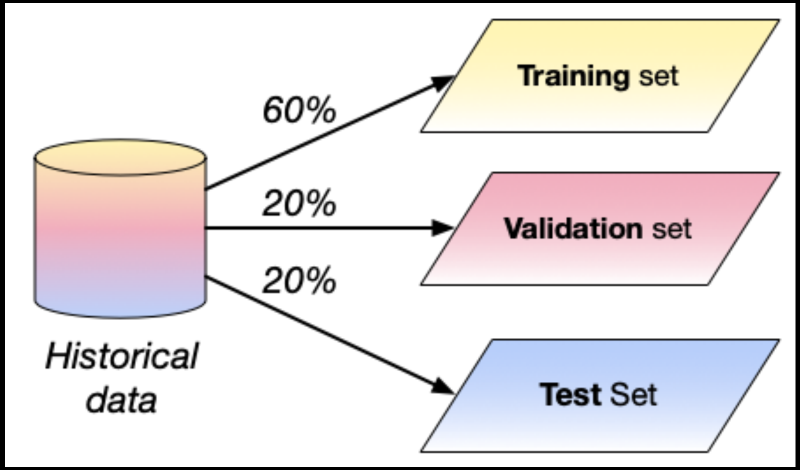


Model Development Lifecycle


Historical data
Training and cross-validation
Training and Validation sets
Model
Validation
Modeling
Test Set
Learned Model
Evaluation
Evaluation
New Data
Developed Model
Prediction
Monitoring
Deployment and Monitoring

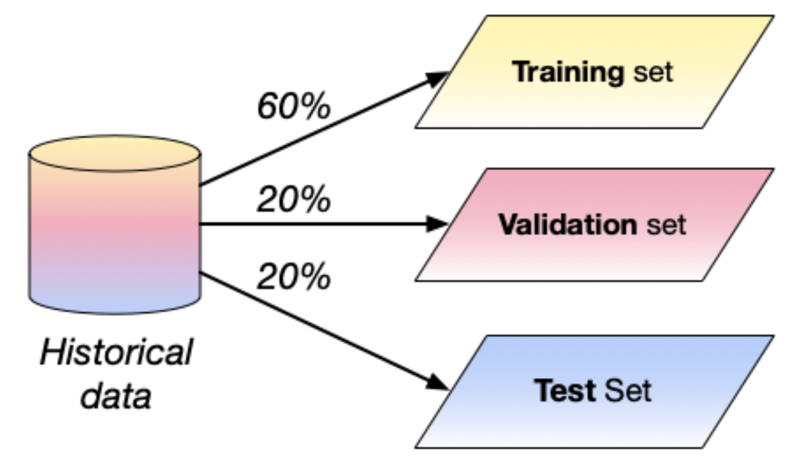
Dataset Splitting

Split your data



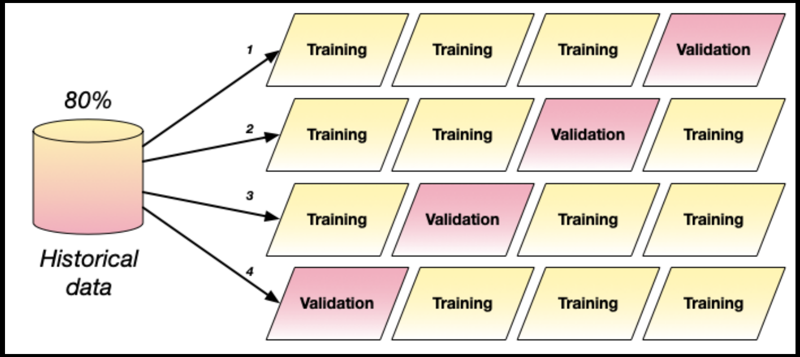
- **Training** set
  - *train*
- **Validation** set
  - *fine-tune*
- **Test** set
  - *evaluate*

Avoid leakages



- **Data items**
  - in the *validation* or *evaluation* sets
- **Features**
  - highly correlated to prediction
  - not present in the production environment

Cross-validation



- Cycle training and validation data several times
  - Useful when dataset is small
- Split the data into n portions
  - Train the model n times using $n-1$ portions for training
  - Average results

Evaluation

How to Evaluate?

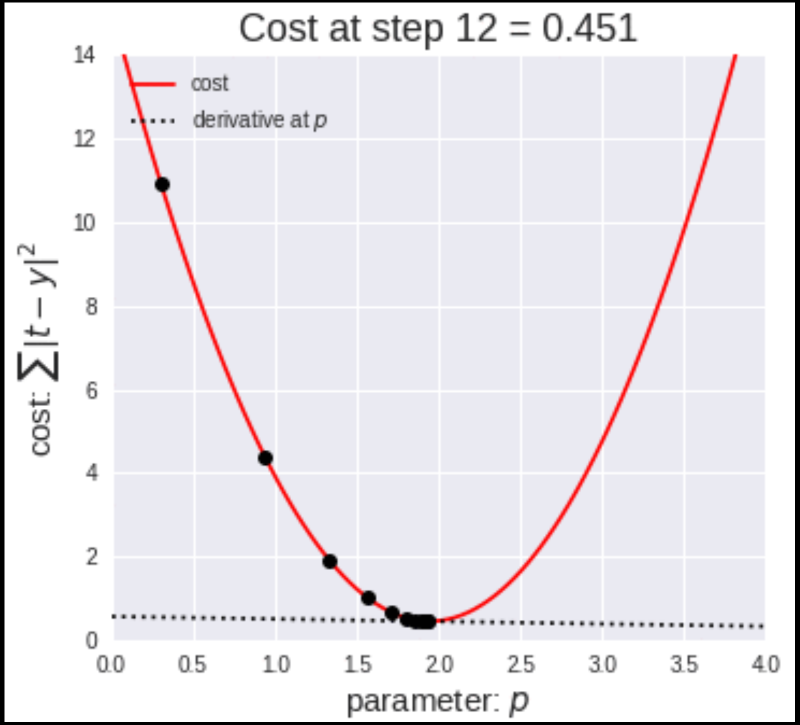
- **Metric**
  - How to measure errors?
  - Both training and testing
- **Training**
  - How to “help” the ML model to perform well?
- **Validation**
  - How to pick the best ML model?
- **Evaluation**
  - How to “help” the ML model to generalize?

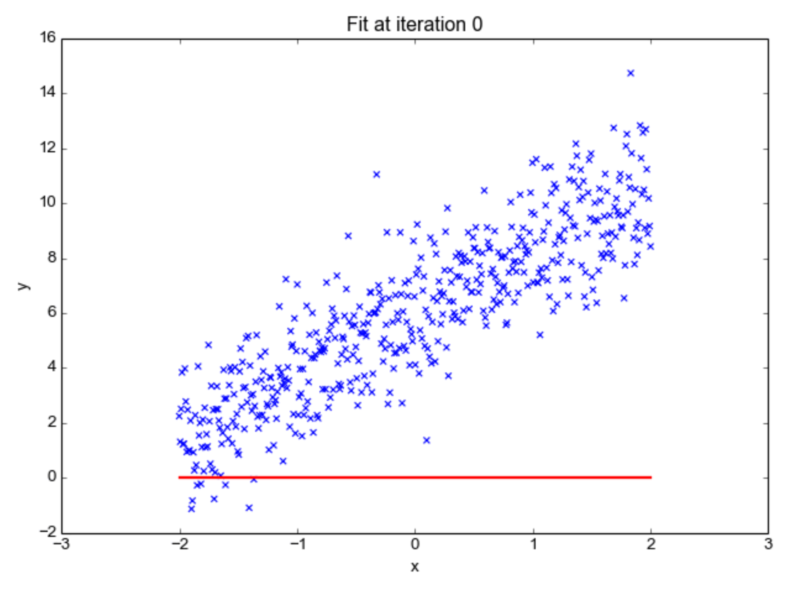
Let errors guide you

- Errors are almost inevitable!

- How to measure errors?
- Select an evaluation procedure (a “metric”)

Model Training Process





Errors

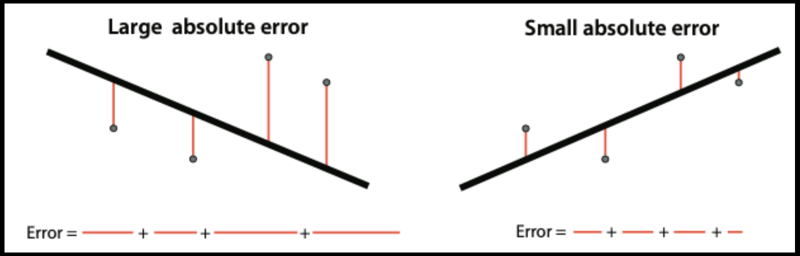
- These are the most common questions:
  - *How* is the prediction wrong?
  - *How often* is the prediction wrong?
  - What is the *cost* of wrong predictions?
  - How does the *cost* vary by the wrong prediction type?
  - How can *costs* be minimised?

Regression

Mean absolute error

$$MAE = \frac{1}{N}\sum_{J=1}^{N}|p_j - v_j|$$

Average of the difference between the *expected* value (v_j) and the *predicted* value (p_j)



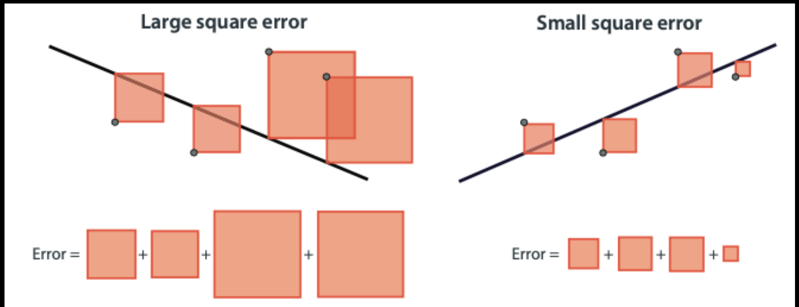
Mean square error

$$MSE = \frac{1}{2N}\sum_{J=1}^{N}(p_j - v_j)^2$$

Average of the square of the difference between the *training* value (v_j) and the *expected* value (p_j)

Square is easier to use during the training process (derivative)

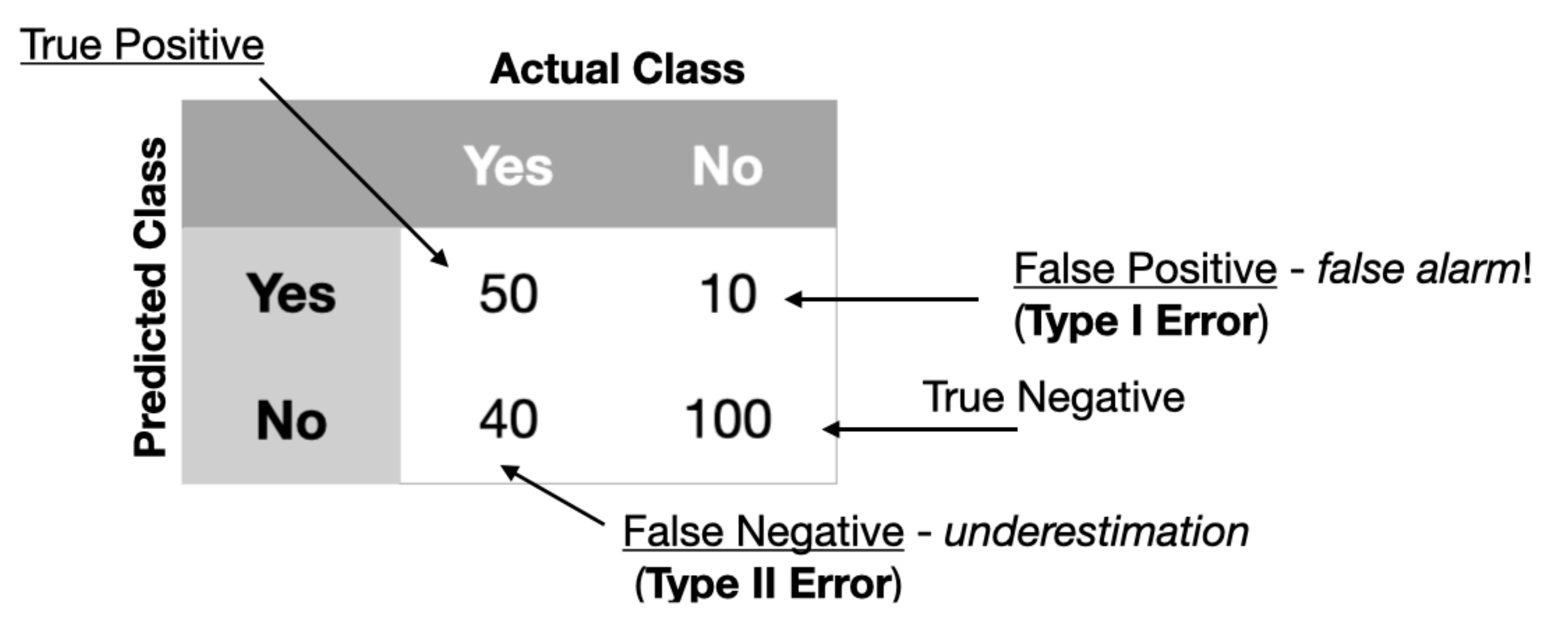
More significant errors are more pronounced



Classification

Confusion Matrix

Describes the complete performance of the model



		Actual Class	
		Yes	No
Predicted Class	Yes	50	10
	No	40	100

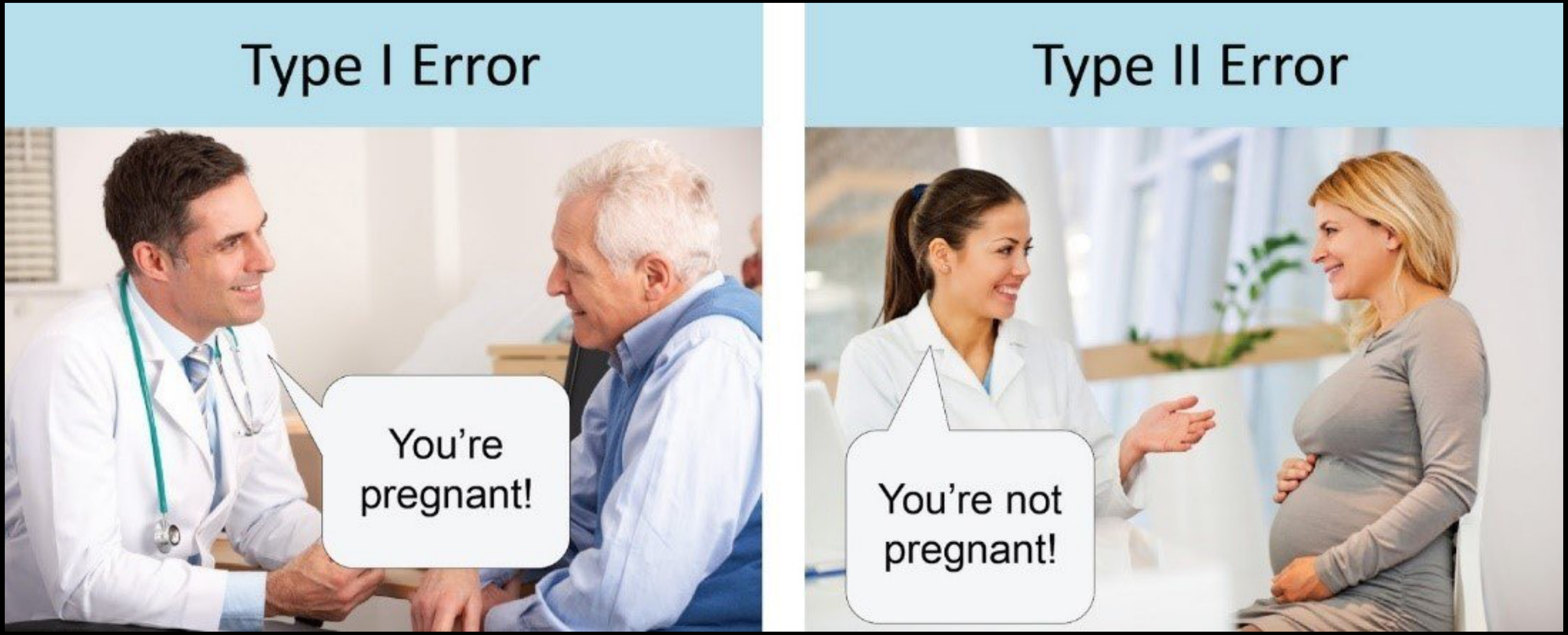
True Positive

False Positive - *false alarm*!
(Type I Error)

True Negative

False Negative - *underestimation*
(Type II Error)

Type I and Type II errors



Accuracy

$$\frac{TP+TN}{TP+TN+FP+FN}$$

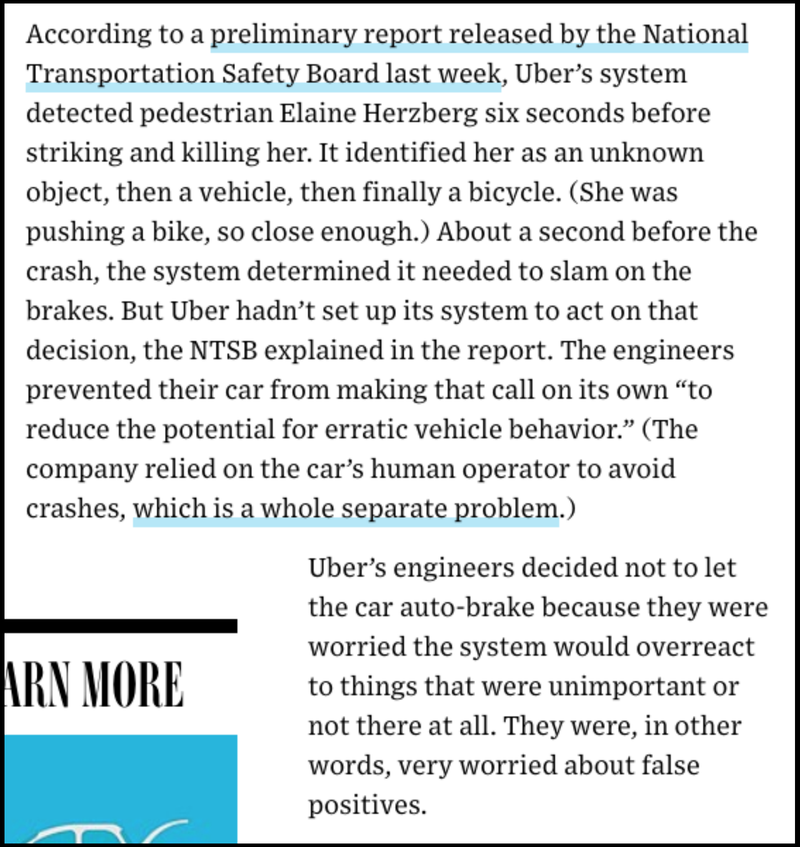
The *percentage of times* that a model is correct

The model with the highest accuracy *is not necessarily the best*

Some errors (e.g., False Negative) can be more costly than others

Errors are not equal

FALSE POSITIVES: SELF-DRIVING CARS AND THE AGONY OF KNOWING WHAT MATTERS



According to a preliminary report released by the National Transportation Safety Board last week, Uber's system detected pedestrian Elaine Herzberg six seconds before striking and killing her. It identified her as an unknown object, then a vehicle, then finally a bicycle. (She was pushing a bike, so close enough.) About a second before the crash, the system determined it needed to slam on the brakes. But Uber hadn't set up its system to act on that decision, the NTSB explained in the report. The engineers prevented their car from making that call on its own "to reduce the potential for erratic vehicle behavior." (The company relied on the car's human operator to avoid crashes, which is a whole separate problem.)

Uber's engineers decided not to let the car auto-brake because they were worried the system would overreact to things that were unimportant or not there at all. They were, in other words, very worried about false positives.

- Detecting the “Alexa” command?
- Pregnancy detection
  - Cost of “false negatives”?
  - Cost of “false positives”?
- Covid testing
  - Cost of “false negatives”?
  - Cost of “false positives”?
- Law enforcement?

Balanced Accuracy

$$\frac{\frac{TP}{TP+FN}+\frac{TN}{FP+TN}}{2}$$

Average of single class performance

Good to use when the distribution of data items in classes is imbalanced

Balanced Accuracy Weighted

$$\frac{\frac{TP}{(TP+FN)*w}+\frac{TN}{(FP+TN)*(1-w)}}{2}$$

Weighted average of single-class performance

Weight depends on the popularity of a class.

Precision

$$\frac{TP}{TP+FP}$$

Among the examples we classified as positive, how many did we correctly classify?

Recall

$$\frac{TP}{TP+FN}$$

Among the positive examples, how many did we correctly classify?

F_1 -Score

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

The harmonic mean between *precision* and *recall*

What is the implicit assumption about the costs of errors?

Sensitivity (true positive rate)

$$\frac{TP}{FN+TP}$$

Identification of the positively labeled data items

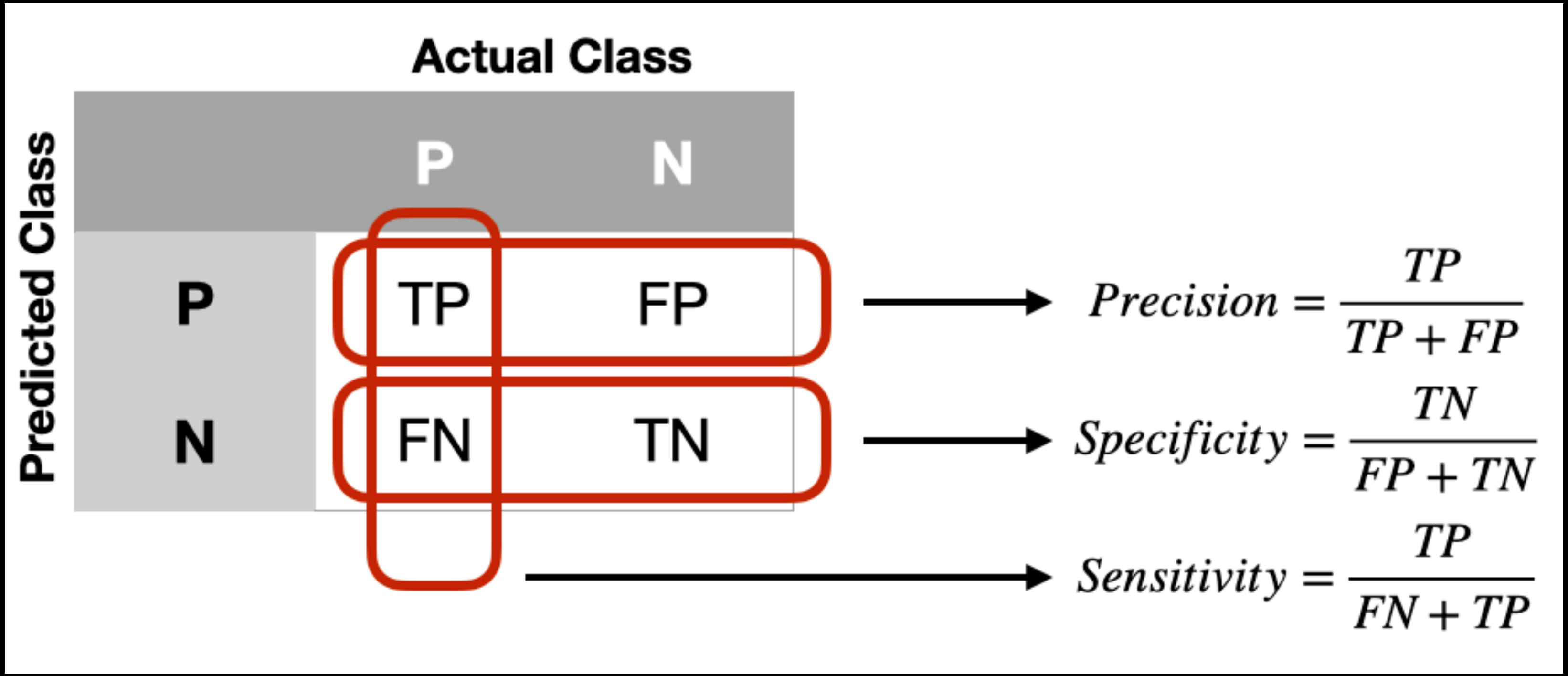
Same as recall

Specificity (false positive rate)

$$\frac{TN}{FP+TN}$$

Identification of the negatively labeled data items

Not the same as precision



Actual Class

Predicted Class

	P	N
P	TP	FP
N	FN	TN

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

$$Specificity = \frac{TN}{FP + TN}$$

$$Sensitivity = \frac{TP}{FN + TP}$$

Medical Test Model

- **Recall** and **sensitivity**
  - How many were correctly diagnosed as sick among the sick people (positives)?
- **Precision**
  - Among the people diagnosed as sick, how many were sick?
- **Specificity**
  - Among the healthy people (negatives), how many were correctly diagnosed as healthy? \u2028

Spam Detection Model

- **Recall** and **sensitivity**
  - How many were correctly deleted among the spam emails (positives)?
- **Precision**
  - Among the deleted emails, how many were spam?
- **Specificity**
  - Among the good emails (negatives), how many were correctly sent to the inbox?\u2028

Search Engine

- Constraint: **high precision**
  - False positives are tolerable but should be minimised
- Among the available models, pick one with a higher recall
  - **Metric**: *Recall at Precision* = $x\%$

Metrics are also designed in a multi-stakeholder context

- One team builds the mode
  - Data scientists / ML engineers
- Many teams will make use of it
  - e.g., product team, management team

Machine Learning for Design

Lecture 5 - Part *a*

Training and Evaluation

Credits

Grokking Machine Learning. Luis G. Serrano. Manning, 2021

[CIS 419/519 Applied Machine Learning]. Eric Eaton, Dinesh Jayaraman.

Deep Learning Patterns and Practices - Andrew Ferlitsch, Maanning, 2021

Machine Learning Design Patterns - Lakshmanan, Robinson, Munn, 2020