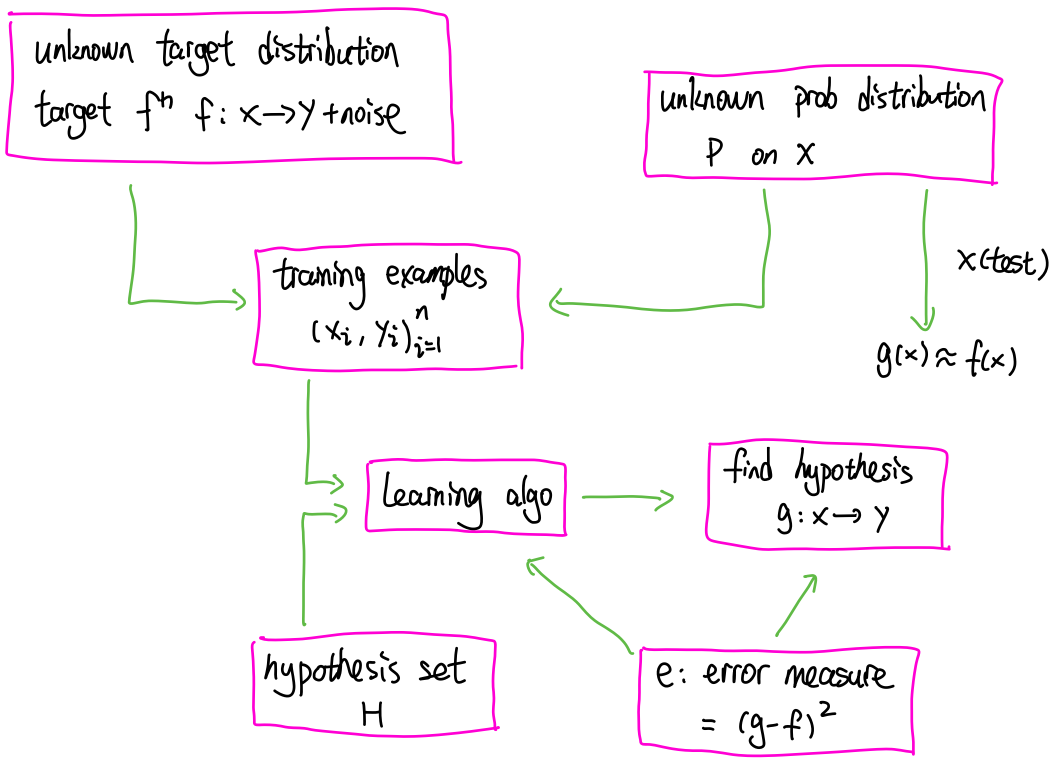
① cross-validation : $f: X \to Y$ Y+noise



⇒ we get a model : how good it is

Dataset
training validation test

Analogy :
lecture, hw, exam
training, val, test

70 training set | 10 val | 20 test
↑
k

need k to be both large and small

⇒ cross-validation comes to rescue

tune hyper-params { 1, 2, 3, 4, . . . k

combine training + validation
split data into k fold
use (k-1) fold for training
1 for validation

↳ Finalize hyper-params
↳ entire training + validation set to select the model

∴ Entire set for training + validation

② Evaluation metrics:

JP Morgan example: detect fraud transaction

$\boxed{x} \to \boxed{M} \to 99\%$ what's wrong?

		Actual	
		+ve	−ve
predicted	+ve	true +ve TP	false +ve FP
	−ve	false −ve FN	true −ve TN

① accuracy : what % of the time modeling is classifying correctly

(when balanced, no skew or class imbalance)

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

② precision : what proportion of predicted positive are truly pos?

(when we want to be very sure of our prediction)

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

③ recall : what proportion of actual pos are correctly predicted?

(when want to capture as many +ve as possible)

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

④ F1 score : trade-off between precision and recall — harmonic mean of the two

$$\frac{1}{F_1} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right) \Rightarrow F_1 = 2\left(\frac{PR}{P+R}\right)$$

$$F_\beta = (1+\beta^2)\,\frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad \text{(weighted)}$$

⑤ Log loss / binary cross entropy :

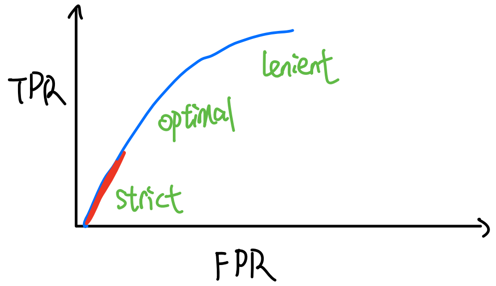
$$L_{BCE} = -\left[y \log p + (1-y)\log(1-p)\right]$$

$p \rightarrow$ prob. of predicting 1

⑥ ROC curve : sensitivity / TPR vs FPR

$$TPR = \text{recall} = \frac{TP}{TP+FN}$$

$$1 - \text{specificity} = FPR = \frac{FP}{TN+FP}$$



AUC: area under the curve

(scale invariant)