Metrics

It is extremely important to use **quantitative metrics** for evaluating a machine learning model

- Until now, we relied on the **cost function value** for regression and classification

- Other metrics can be used to **better evaluate** and understand the model

- **<u>For classification</u>**
  - ✓ Accuracy/Precision/Recall/F1-score, ROC curves,…
- **<u>For regression</u>**
  - ✓ Normalized RMSE, Normalized Mean Absolute Error (NMAE),…

Accuracy

Accuracy is a measure of **how close** a given set of guessing from our model are closed to their true value.

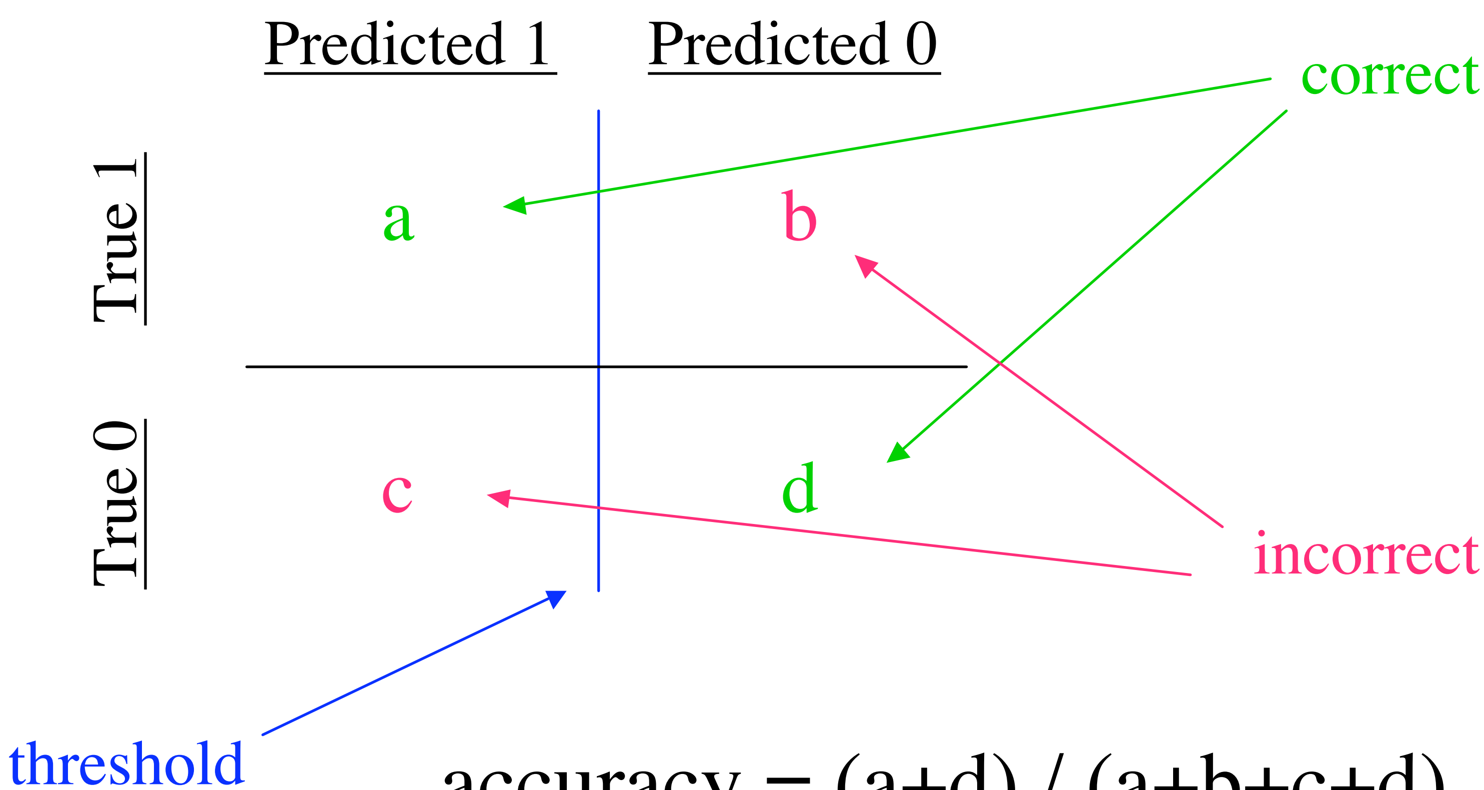
$$\text{Accuracy} = \frac{\#\ \text{Correct classifications}}{\#\ \text{All classifications}}$$

- If a classifier make 10 predictions and 9 of them are correct, the accuracy is 90%.

➢ Accuracy is a measure of **how well** a binary classifier correctly identifies or excludes a condition.

➢ It's the **proportion of correct predictions among the total number of cases examined.**

Confusion Matrix



accuracy = (a+d) / (a+b+c+d)

Classification case: metrics for skewed classes

Disease dichotomic classification example

Train logistic regression model $h(\boldsymbol{x})$, with $y = 1$ if disease, $y = 0$ otherwise.

Find that you got 1% error on test set (99% correct diagnoses)

Only 0.5% of patients **actually have** disease

The $y = 1$ class has very few samples with respect to the $y = 0$ class

If I use a classifier that **always classifies** the observations to the **0 class**, I get 99.5% of accuracy!!

For **skewed classes,** the accuracy metric can be deceptive

Precision and recall

Suppose that $y = 1$ in presence of a **rare class** that we want to detect

Precision *(How much we are precise in the detection)*

Of all patients where we classified $y = 1$,
what fraction actually has the disease?

$$\frac{\text{True Positive}}{\text{\# Estimated Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall *(How much we are good at detecting)*

Of all patients that actually have the disease, what fraction did we correctly detect as having the disease?

$$\frac{\text{True Positive}}{\text{\# Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Confusion matrix

Actual class

Estiamted class	1 (p)	0 (n)
1 (Y)	True positive (TP)	False positive (FP)
0 (N)	False negative (FN)	True negative (TN)

Trading off precision and recall

Logistic regression: $0 \leq s(\boldsymbol{\varphi}^{\top}\boldsymbol{\theta}) \leq 1$

- Classify 1 if $s(\boldsymbol{\varphi}^{\top}\boldsymbol{\theta}) \geq 0.5$
- Classify 0 if $s(\boldsymbol{\varphi}^{\top}\boldsymbol{\theta}) < 0.5$

These thresholds can be different from 0.5!

→ *At different thresholds, correspond different confusion matrices!*

Suppose we want to classify $y = 1$ (disease) only if very confident

- Increase threshold → Higher precision, lower recall

Suppose we want to avoid missing too many cases of disease (avoid false negatives)

- Decrease threshold → Higher recall, lower precision

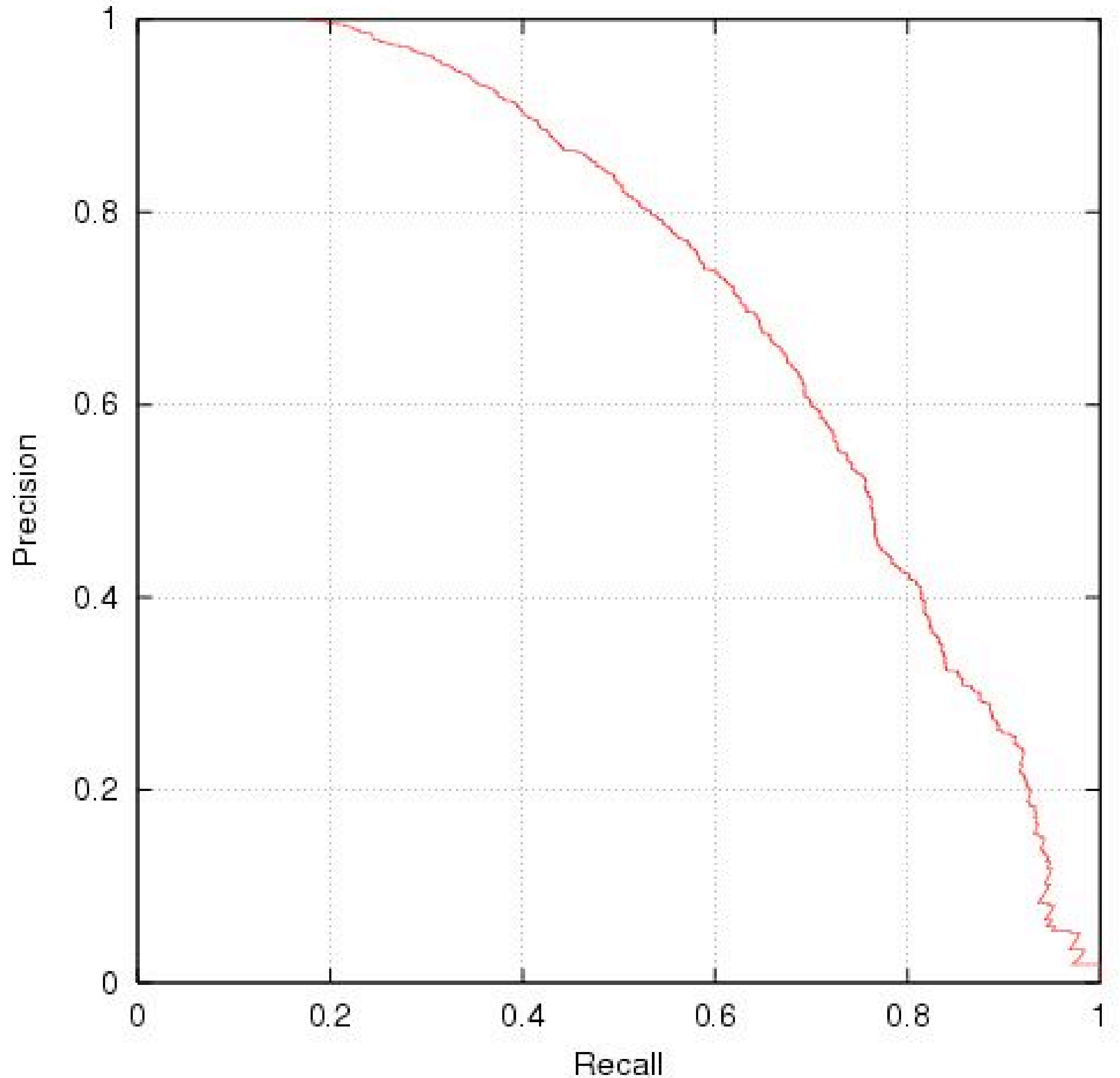
Precision/Recall

	Predicted 1	Predicted 0
True 1	a	b
True 0	c	d

threshold

$$PRECISION = a/(a+c)$$

$$RECALL = a/(a+b)$$


1
0.8
0.6
0.4
0.2
0
Precision
0
0.2
0.4
0.6
0.8
1
Recall

F1-score

It is usually better to compare models by means of one number only. The $\mathbf{F1-score}$ can be used to **combine precision and recall**

	Precision(P)	Recall (R)	Average	F_1 Score
Algorithm 1	0.5	0.4	0.45	$\boxed{0.444}$
Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.51	0.0392

The best is Algorithm 1

Algorithm 3 classifies always 1

Average says not correctly that Algorithm 3 is the best

$$\mathrm{Average} = \frac{\mathrm{P} + \mathrm{R}}{2} \qquad \mathrm{F_1score} = 2\frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

- $\mathrm{P} = 0 \text{ or } \mathrm{R} = 0 \Rightarrow \mathrm{F_1score} = 0$
- $\mathrm{P} = 1 \text{ and } \mathrm{R} = 1 \Rightarrow \mathrm{F_1score} = 1$

Summaries of the confusion matrix

Different metrics can be computed from the confusion matrix, depending on the class of interest *(https://en.wikipedia.org/wiki/Precision_and_recall)*

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $\frac{1}{\frac{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}{2}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $\boldsymbol{CM_{i,j}}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%