UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Lesson 6.

Performance metrics

DYNAMIC SYSTEMS IDENTIFICATION COURSE

MASTER DEGREE ENGINEERING AND MANAGEMENT FOR HEALTH

TEACHER
Mirko Mazzoleni

PLACE
University of Bergamo

Outline

1. Metrics

2. Precision and recall

3. Receiver Operating Characteristic (ROC) curves

4. Worked example

Outline

1. **Metrics**

2. Precision and recall

3. Receiver Operating Characteristic (ROC) curves

4. Worked example

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

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

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 examples with respect to the $y = 0$ class

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

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

Outline

1. Metrics

2. **Precision and recall**

3. Receiver Operating Characteristic (ROC) curves

4. Worked example

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	
		1 (p)	**0 (n)**
Estiamted class	**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

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	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

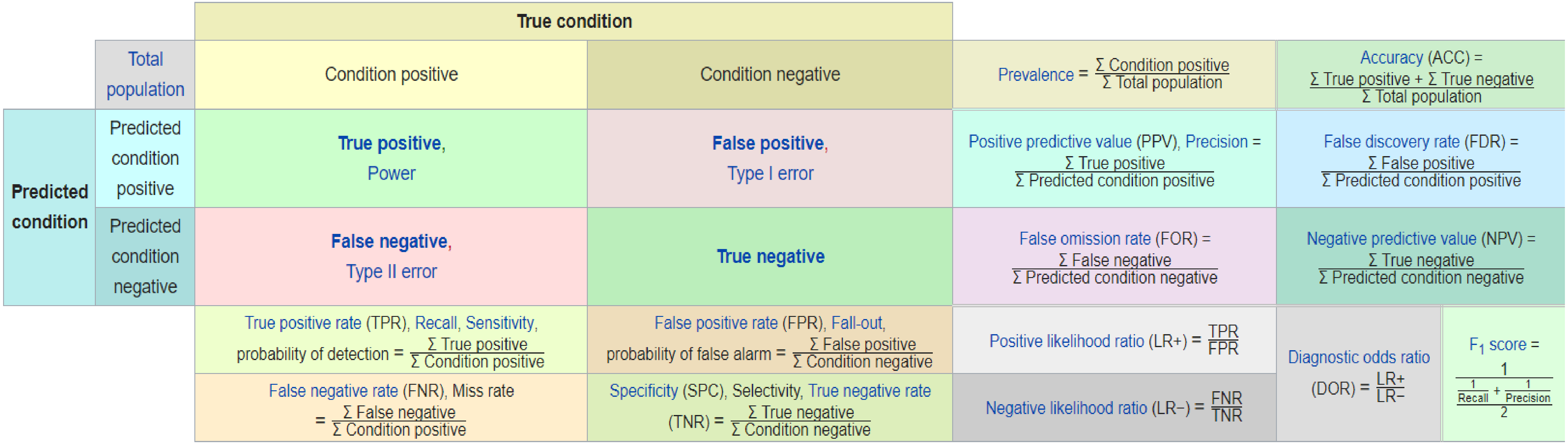
Average says not correctly that Algorithm 3 is the best

$$\text{Average} = \frac{\text{P} + \text{R}}{2} \qquad \text{F}_1\text{score} = 2\,\frac{\text{P} \cdot \text{R}}{\text{P} + \text{R}}$$

- $\text{P} = 0 \text{ or } \text{R} = 0 \Rightarrow \text{F}_1\text{score} = 0$
- $\text{P} = 1 \text{ and } \text{R} = 1 \Rightarrow \text{F}_1\text{score} = 1$

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)*



<table>
<tr><td></td><td></td><td colspan="2"><b>True condition</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2"><b>Predicted condition</b></td><td>Predicted condition positive</td><td><b>True positive</b>,
Power</td><td><b>False positive</b>,
Type I error</td><td>Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td><b>False negative</b>,
Type II error</td><td><b>True negative</b></td><td>False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td></td><td></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$\text{F}_1$ score = $\frac{1}{\frac{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}{2}}$</td></tr>
<tr><td></td><td></td><td>False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

Outline

1. Metrics

2. Precision and recall

3. **Receiver Operating Characteristic (ROC) curves**

4. Worked example

Ranking instead of classifying

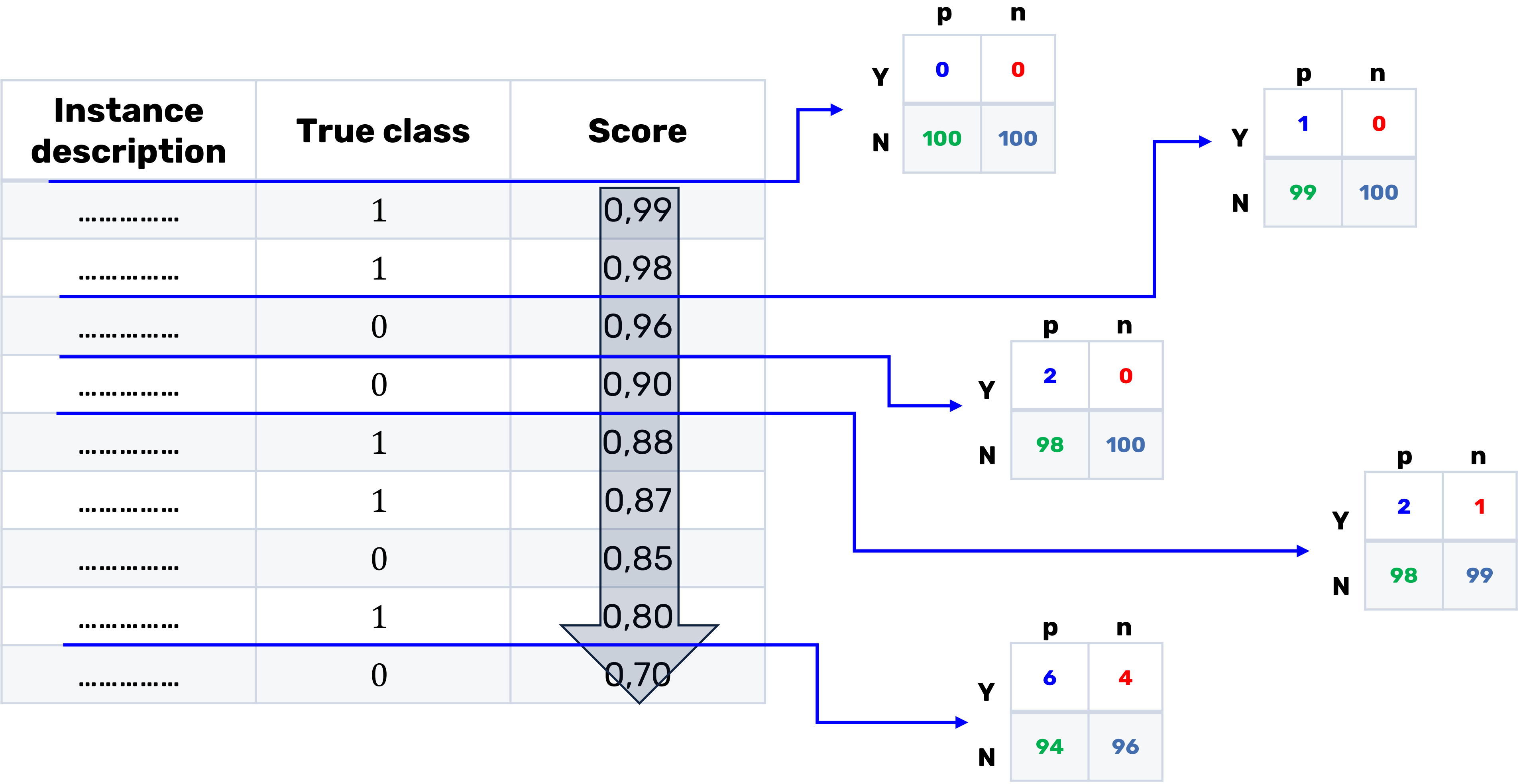
Classifiers such as logistic regression can output a **probability** of belonging to a class (or something similar)

- We can use this to **rank** the different istances and take actions on the cases at top of the list

- We may have a **budget**, so we have to target most promising individuals

- Ranking enables to use different techniques for **visualizing** model performance

Ranking instead of classifying



Instance description	True class	Score
……………	1	0,99
……………	1	0,98
……………	0	0,96
……………	0	0,90
……………	1	0,88
……………	1	0,87
……………	0	0,85
……………	1	0,80
……………	0	0,70

$$\begin{array}{c|cc} & p & n \\ \hline Y & 0 & 0 \\ N & 100 & 100 \end{array}$$

$$\begin{array}{c|cc} & p & n \\ \hline Y & 1 & 0 \\ N & 99 & 100 \end{array}$$

$$\begin{array}{c|cc} & p & n \\ \hline Y & 2 & 0 \\ N & 98 & 100 \end{array}$$

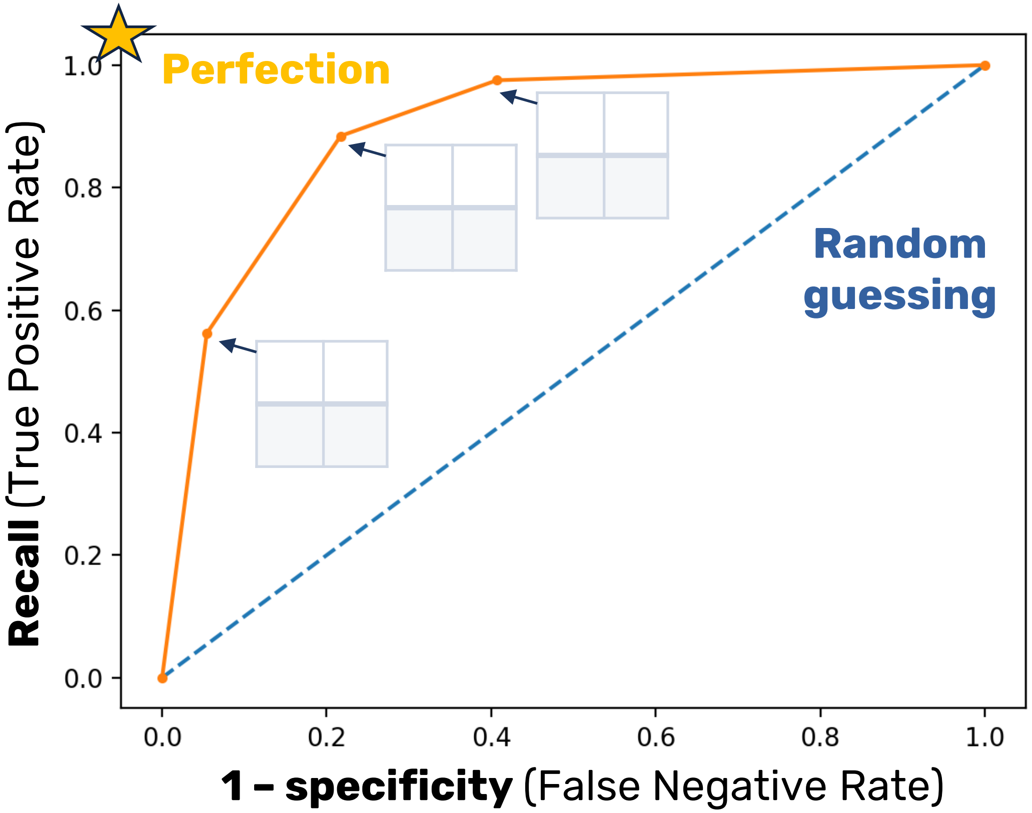
$$\begin{array}{c|cc} & p & n \\ \hline Y & 2 & 1 \\ N & 98 & 99 \end{array}$$

$$\begin{array}{c|cc} & p & n \\ \hline Y & 6 & 4 \\ N & 94 & 96 \end{array}$$

Different confusion matrices by changing the **threshold**

Ranking instead of classifying

ROC curves are a very general way to **represent and compare** the performance of different models (on a binary classification task)



Observations

- (0,0): classify always negative
- (1,1): classify always positive
- Diagonal line: random classifier
- Below diagonal line: worse than random classifier
- Different classifiers can be compared
- Area Under the Curve (AUC): probability that a randomly chosen positive instance will be ranked ahead of randomly chosen negative instance

Outline

1. Metrics

2. Precision and recall

3. Receiver Operating Characteristic (ROC) curves

4. **Worked example**

Disclaimer

This example is **ONLY** for **educational purposes**, in order to see how to train and use a convolutional neural network in practice with real data.

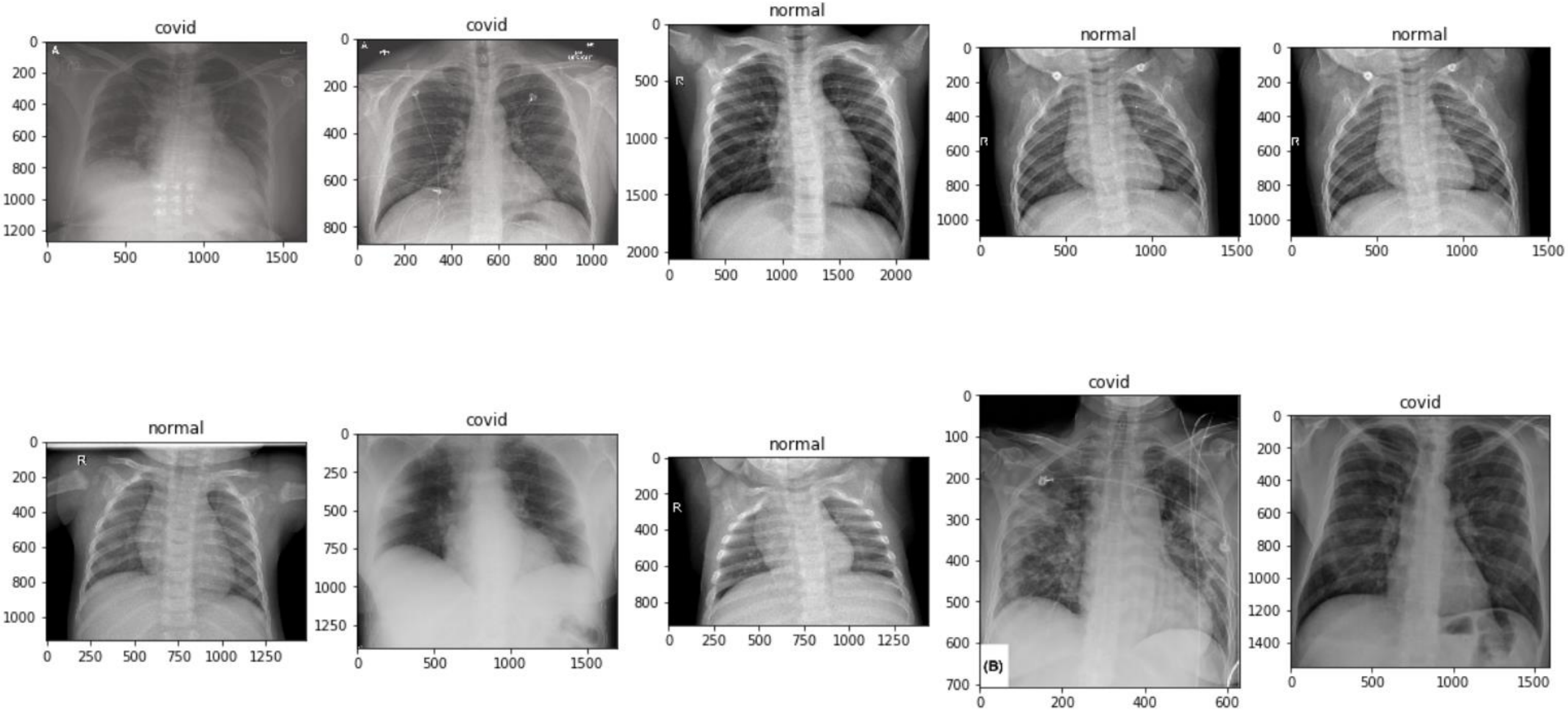
I am **NOT**, by any means, trying to say that this should be an accurate or valid system from a medical point of view.

Artificial intelligence tools show **ALWAYS be supported** by domain knowledge from humans.

Again, this example does not claim to solve COVID-19 detection.

Pneumonia detection

Suppose to have at disposal X-ray images of lungs: **Healthy** people - **Covid-19 disease** patients



This example is **ONLY** for **educational purposes**

Acknowledgments

- The COVID-19 X-ray image is curated by Dr. Joseph Cohen, a postdoctoral fellow at the University of Montreal, see https://josephpcohen.com/w/public-covid19-dataset/

- The previous data contain only X-ray images of people with a disease. To collect images of healthy people, we can download another X-ray dataset on the platform Kaggle https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

- The analysis is inspired from a tutorial by Adrian Rosebrock: https://www.pyimagesearch.com/2020/03/16/detecting-covid-19-in-x-ray-images-with-keras-tensorflow-and-deep-learning/

This example is **ONLY** for **educational purposes**

Acknowledgments

We want to use a classifier to perform classification:

- **Healthy** patients: class 0
- Patients with a **disease**: class 1

The input data are directly the X-ray **images**

For these computer vision tasks, the state of the art algorithm are the **Convolutional Neural Networks:**

- we can use them to classify the images into **healthy** and **disease**

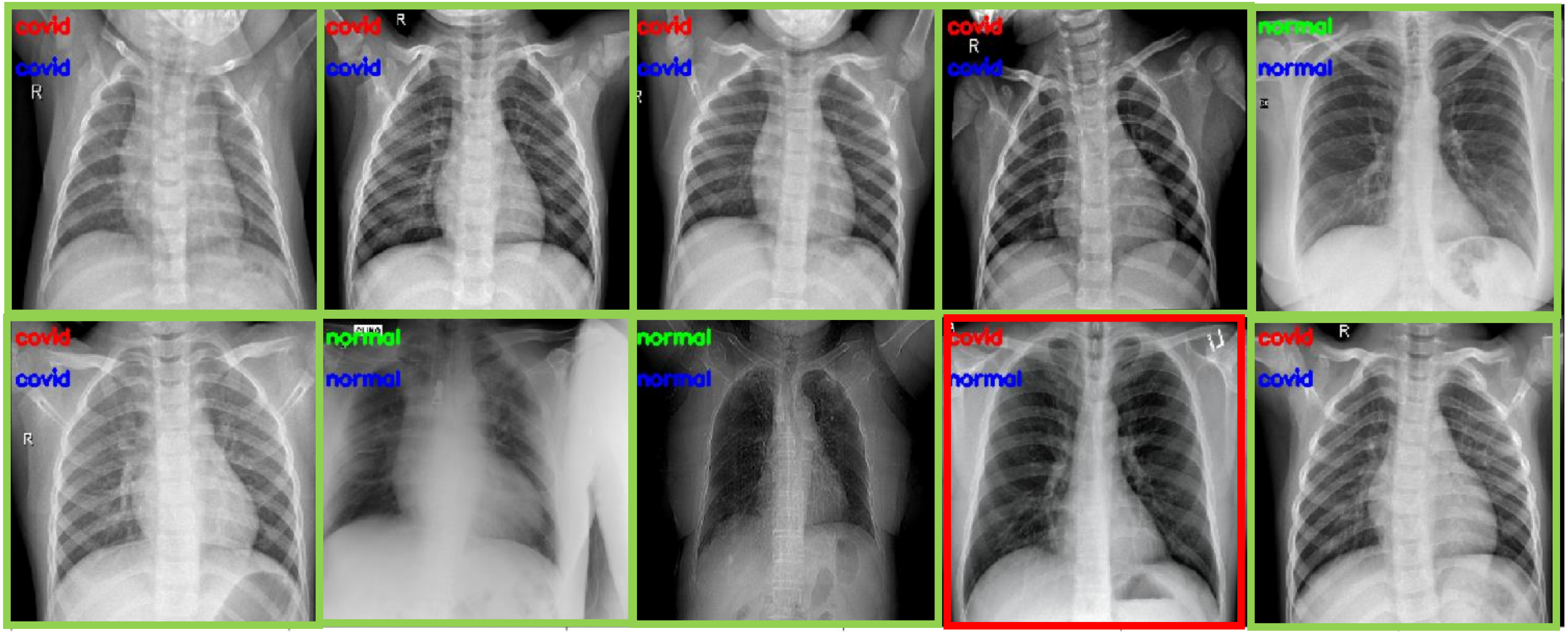
This example is **ONLY** for **educational purposes**

Pneumonia detection

True label

Estimated covid label

Estimated healthy label



This example is **ONLY** for **educational purposes**

Pneumonia detection

Classification results on test set

Sensitivity (recall, true positive rate)

$$\frac{\text{True Positive}}{\text{\# Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = 0.92$$

Specificity (true negative rate)

$$\frac{\text{True Negative}}{\text{\# Actual Negative}} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} = 1$$

Actual class

Estimated class

	1 (p)	**0 (n)**
1 (Y)	**True positive** **11**	**False positive** **0**
0 (N)	**False negative** **1**	**True negative** **11**

- **Accuracy**: $\approx 96\%$

This example is **ONLY** for **educational purposes**

Pneumonia detection

Classification results on test set

Sensitivity (recall, true positive rate)

$$\frac{\text{True Positive}}{\text{\# Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = 0.92$$

Specificity (true negative rate)

$$\frac{\text{True Negative}}{\text{\# Actual Negative}} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} = 1$$

- **Sensitivity:** of patients that ***do have*** COVID-19 (i.e., *true positives*), we could accurately identify them as "COVID-19 positive" 92% of the time using our model

- **Specificity:** of patients that ***do not have*** COVID-19 (i.e., true negatives), we could accurately identify them as "COVID-19 negative" 100% of the time using our model.

This example is **ONLY** for **educational purposes**

Pneumonia detection

Classification results on test set

Sensitivity (recall, true positive rate)

$$\frac{\text{True Positive}}{\text{\# Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = 0.92$$

Specificity (true negative rate)

$$\frac{\text{True Negative}}{\text{\# Actual Negative}} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} = 1$$

- Being able to **accurately detect healthy patients** with 100% accuracy is great. We do not want to quarantine someone for nothing

- …but **we don’t want to classify someone as «healthy» when they are «COVID-19 positive»**, since it could infect other people without knowing

This example is **ONLY** for **educational purposes**

Summary

Balancing sensitivity and specificity is incredibly challenging when it comes to medical applications

The results should **always be validated** with another pool of people

Furthermore, we need to be **concerned of what the model is actually learning:**

- Does the results align with the medical knowledge?
- Was the dataset well representative of the population or there was selection bias?
- Do we accounted for all external factors (confounding) that could interfere with the response?

UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione