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
Antonio Ferramosca

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),…

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.**

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

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

Estiamted class \ Actual 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

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

$$\text{Average} = \frac{P + R}{2} \qquad F_1\text{score} = 2\frac{P \cdot R}{P + R}$$

- $P = 0$ or $R = 0 \Rightarrow F_1\text{score} = 0$
- $P = 1$ and $R = 1 \Rightarrow F_1\text{score} = 1$

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)*

		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 = $\frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}{2}$
		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−) = $\frac{\text{FNR}}{\text{TNR}}$		

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

	p	n
Y	0	0
N	100	100

	p	n
Y	1	0
N	99	100

	p	n
Y	2	0
N	98	100

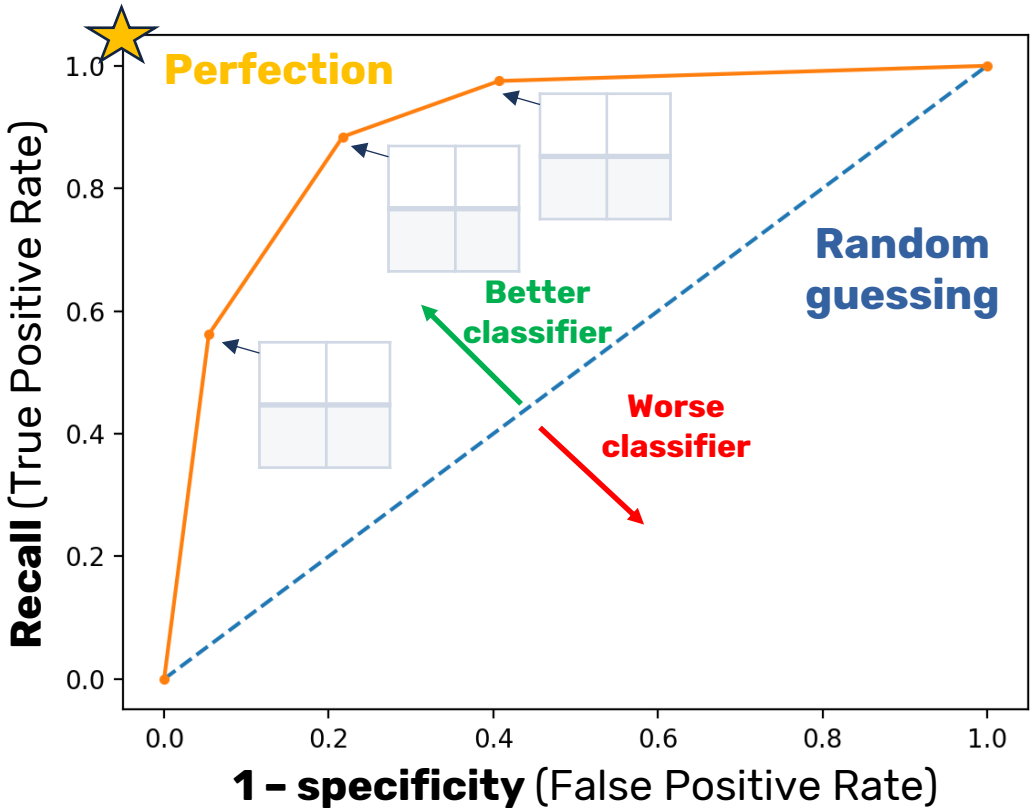
	p	n
Y	2	1
N	98	99

	p	n
Y	6	4
N	94	96

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 examples**

Breast cancer detection

- Breast cancer is the most common cancer amongst women in the world.
- It accounts for 25% of all cancer cases, and affected over 2.1 Million people in 2015 alone.
- It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area.
- The key challenges against it's detection is how to classify tumors into malignant (cancerous) or benign(non cancerous).
- **Goal:** classifying these tumors using machine learning and the Breast Cancer Wisconsin (Diagnostic) Dataset.

Breast cancer Wisconsin dataset

This dataset has been referred from Kaggle.

Output:
Class 4 stands for malignant cancer
Class 2 stands for benign cancer

id_num	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1041801	5	3	3	3	2	3	4	4	1	4
1043999	1	1	1	1	2	3	3	1	1	2
1044572	8	7	5	10	7	9	5	5	4	4
1047630	7	4	6	4	6	1	4	3	1	4
1048672	4	1	1	1	2	1	2	1	1	2
1049815	4	1	1	1	2	1	3	1	1	2
1050670	10	7	7	6	4	10	4	1	2	4
……	……	……	……	……	……	……	……	…….	…….	…….

Breast cancer detection

We will use the dataset to compare differente logistic regression models by means of the ROC curve associated to each of them.

To this aim we will work with 4 different dataset (plus an extra one)

1. **Case 1:** the whole dataset
2. **Case 2:** the first group of 5 features
3. **Case 3:** the second group of 5 features
4. **Case 4:** only the first two features

Extra: after learning the model of CASE 1, take only the features with the smallest p-value.

Matlab code

Output:

- Class 4 stands for malignant cancer and it is for us the positive output. We set it to 1
- Class 2 stands for benign cancer and it is for us the negative output. We set it to 0.

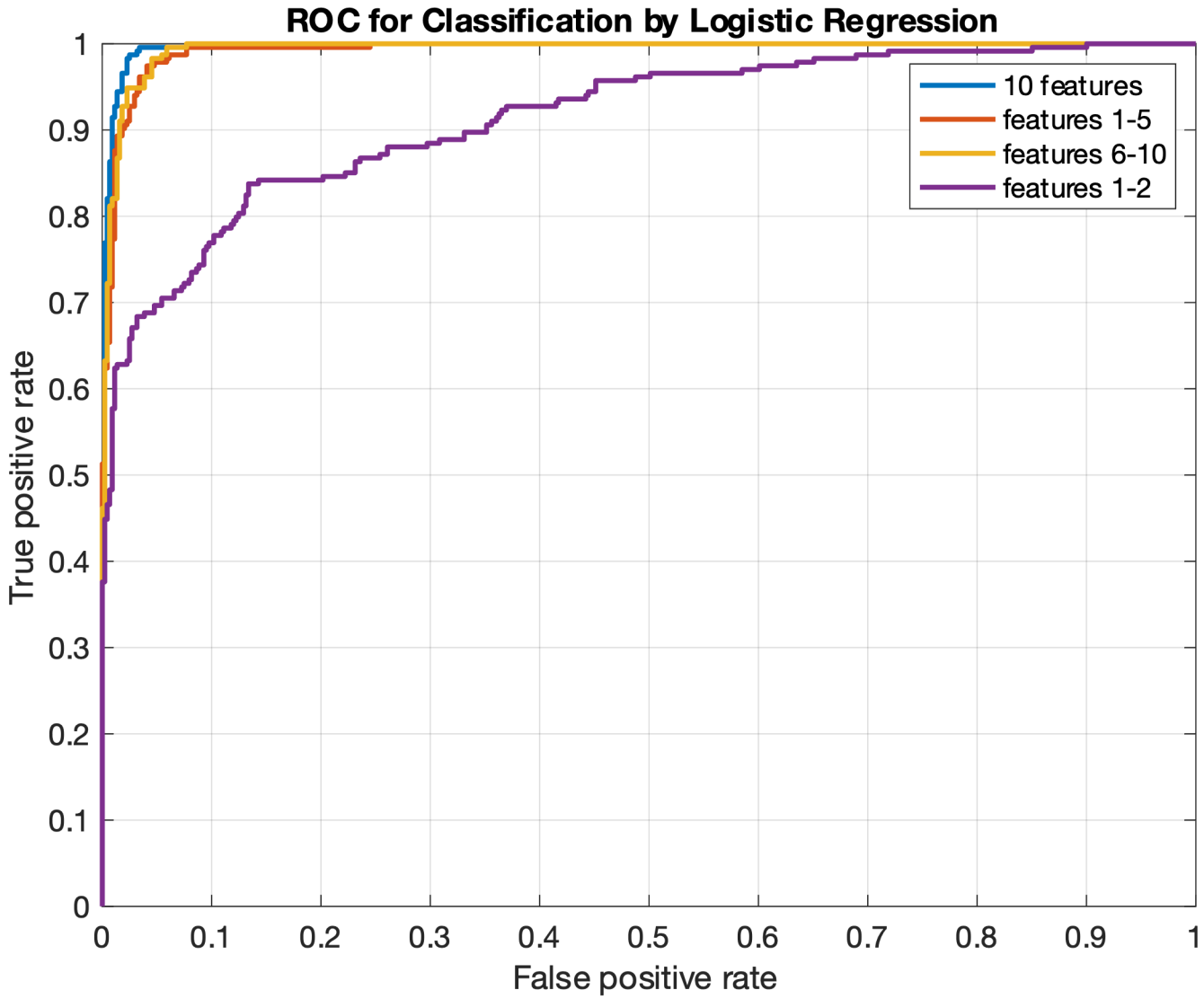
`perfcurve` compute the points in the ROC curve as well as the AUC

```
%% Load and clean data

data = readtable('breast_cancer_w.xlsx'); %load our data as a table

Phi=table2array(data(:,1:end-1));
y=table2array(data(:,end));

y(y==4)=1; % in the original date 4 stands for malignant cancer
y(y==2)=0; % in the original date 2 stands for benign cancer

% Setup the data matrix appropriately, and add ones for the intercept
term
[N, d] = size(Phi);


Phi = [ones(N, 1) Phi]; % Add intercept term
%% Train and test data

mdl = fitglm(Phi,y,'Distribution','binomial','Link','logit')

%% ============ Part 2: Compute the ROC curve ============
scores = mdl.Fitted.Probability;

[X,Y,T,AUC] = perfcurve(y,scores,1);

%Plot the ROC curve.
figure
plot(X,Y)
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC for Classification by Logistic Regression')
```

Results

Comparison of case 1, 2, 3 and 4

Using only the first 2 features is not a smart choice.



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

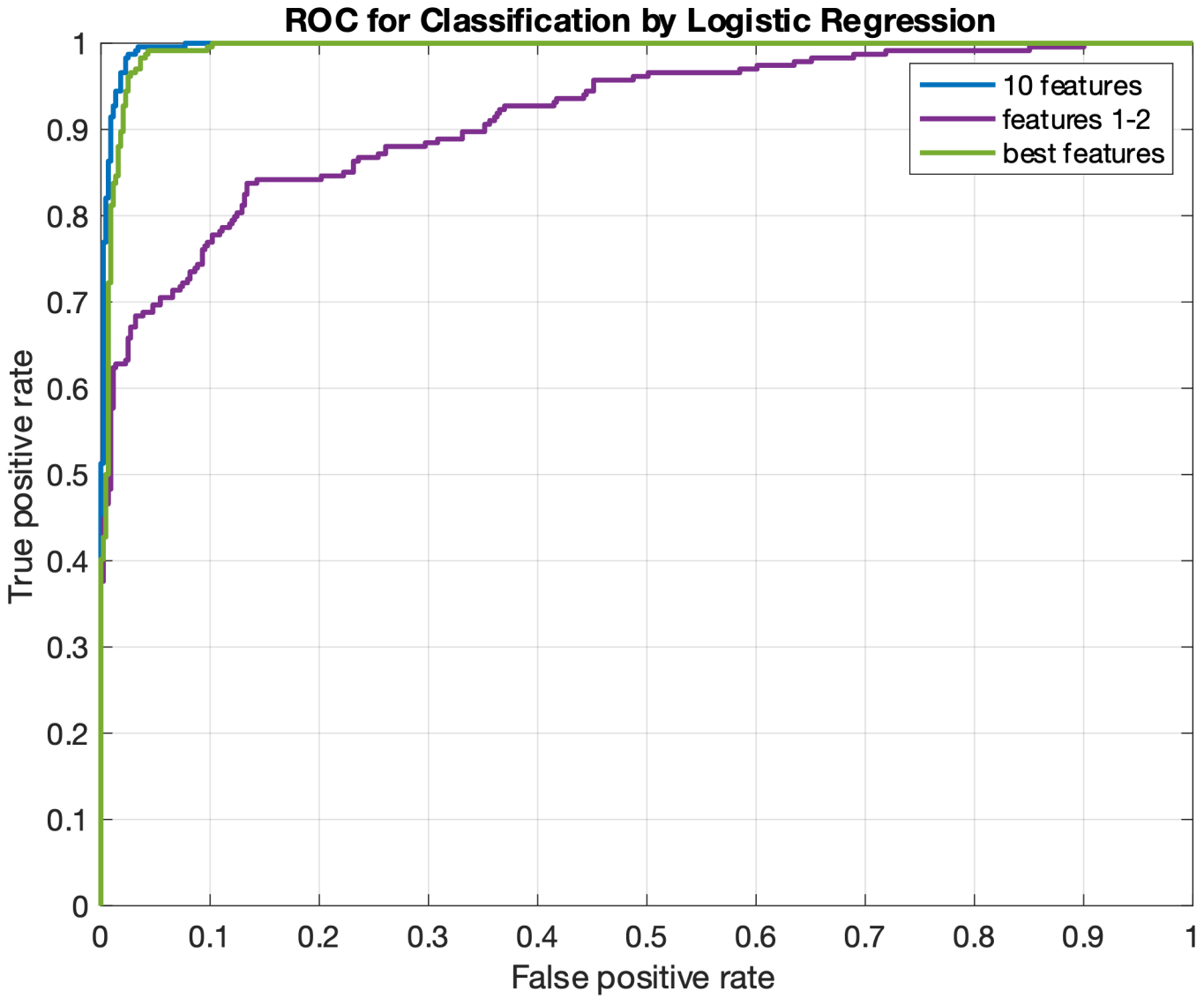
Results

Comparison of case 1, 4 and best

Using only the best features provides a model that performs almost as well as using all the features

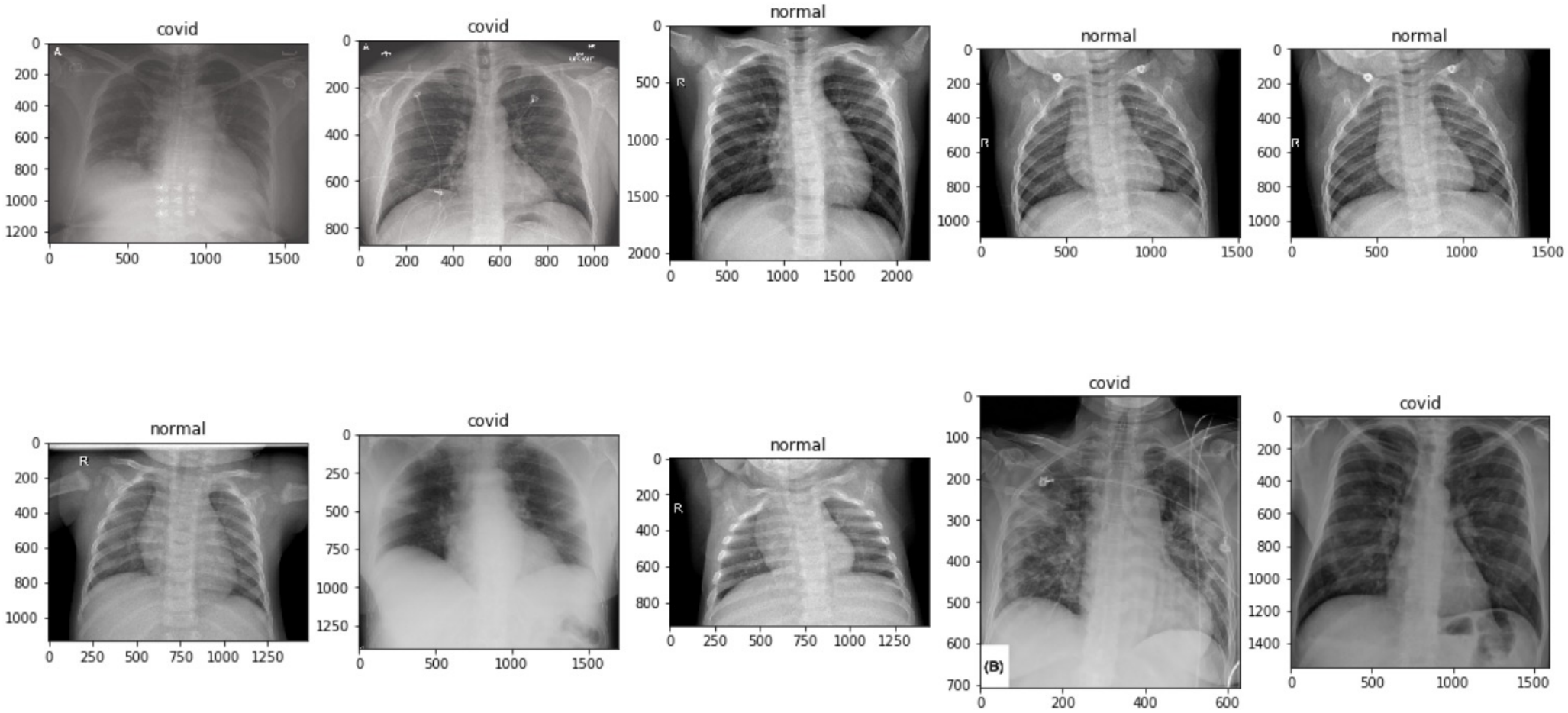
Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-10.349	1.2285	-8.424	3.6385e-17
x1	0	0	NaN	NaN
x2	0.53037	0.14708	3.606	0.00031094
x3	-0.055308	0.21912	-0.25241	0.80072
x4	0.33227	0.23628	1.4062	0.15966
x5	0.34678	0.12727	2.7248	0.0064347
x6	0.1186	0.15955	0.74334	0.45728
x7	0.38473	0.099346	3.8726	0.00010767
x8	0.53256	0.1843	2.8897	0.0038566
x9	0.24116	0.12137	1.987	0.046924
x10	0.52527	0.34241	1.5341	0.12502



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/

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:**

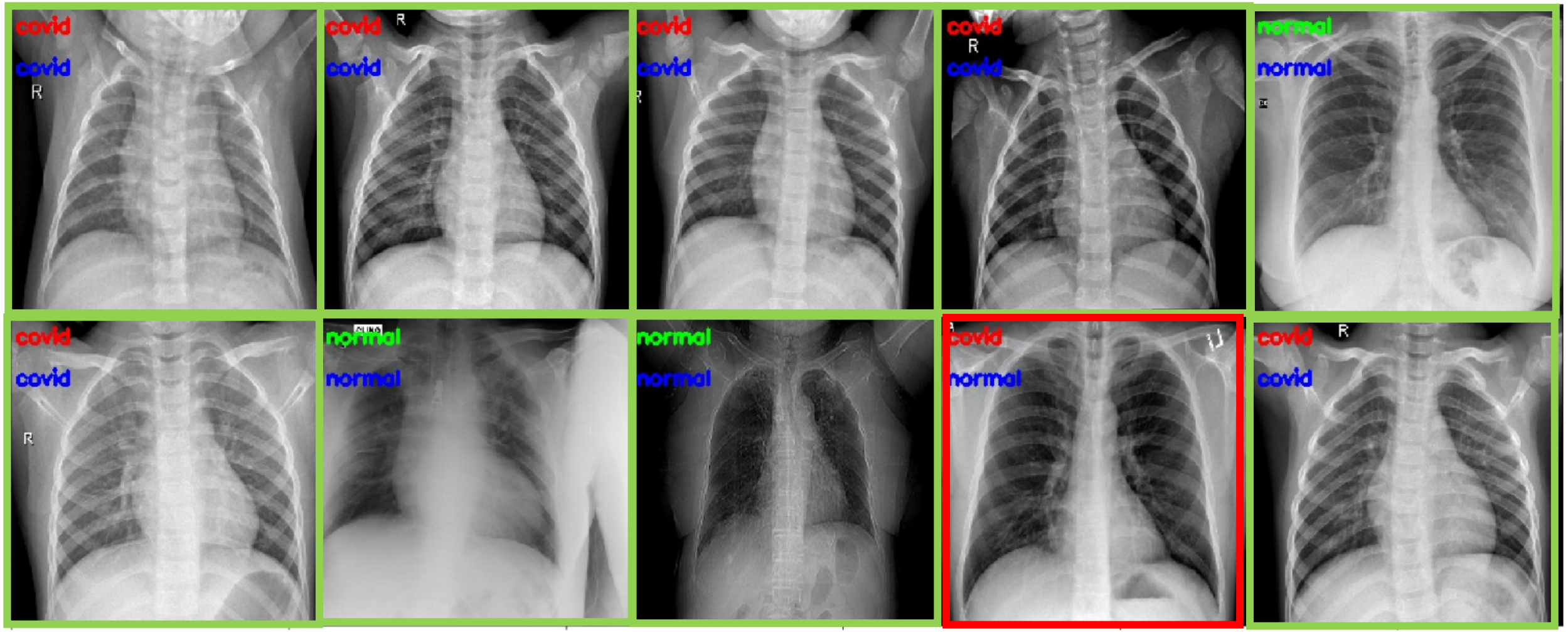
- we can use them to classify the images into **healthy** and **disease**

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)

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

Estimated class	Actual class: 1 (p)	Actual class: 0 (n)
1 (Y)	True positive 11	False positive 0
0 (N)	False negative 1	True negative 11

- **Accuracy**: ≈ 96%

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.

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

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

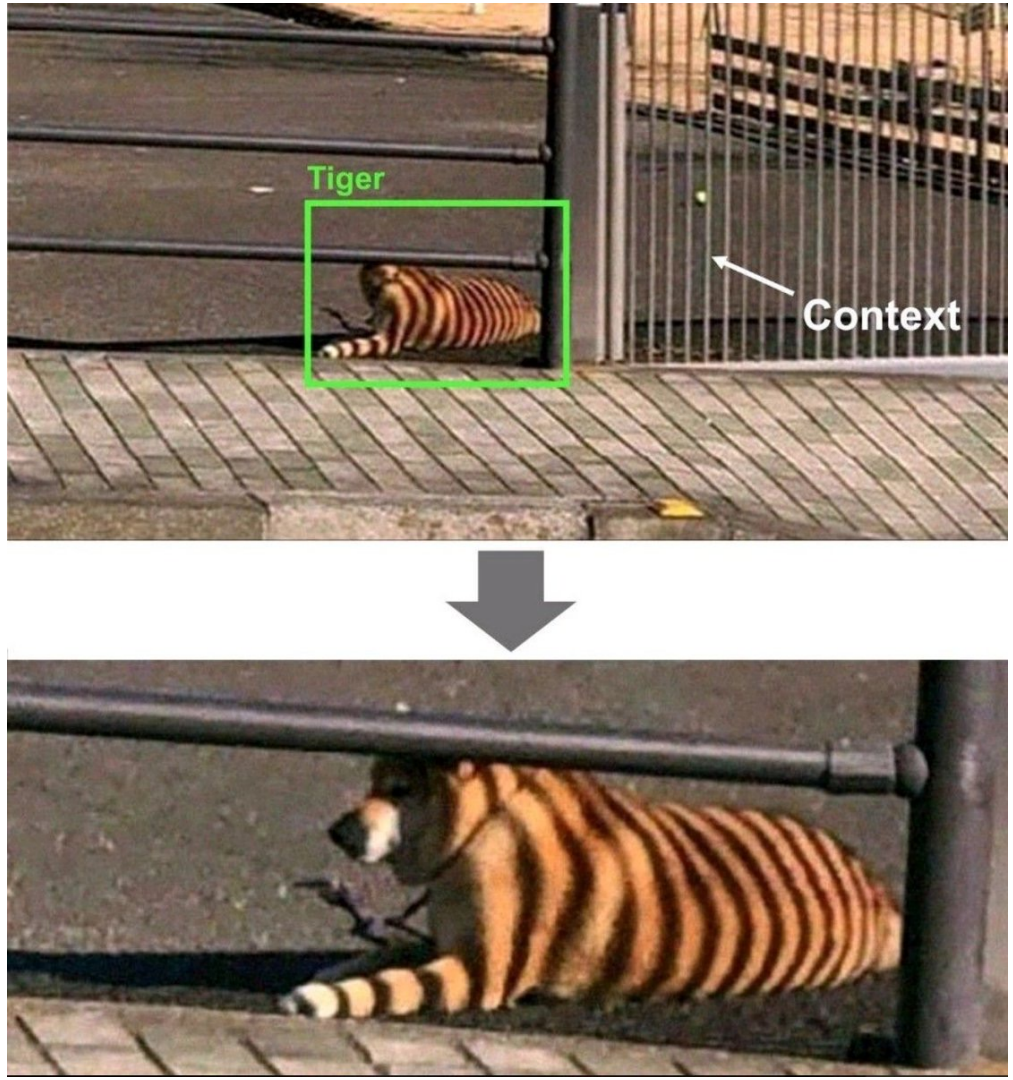
Furthermore, we need to be **concerned of what the model is actually learning:**

- Does the results align with the medical knowledge?
- Was the dataset well representative of the population or there was selection bias?

Summary

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

- 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