UNIVERSITY OF WARWICK

Lecture 8

Evaluating and Comparing Models: Multiclass Classification and Error Analysis

Gabriele Pergola and Amina Asif

Machine Learning Algorithms and Practice
CS429

October 2026

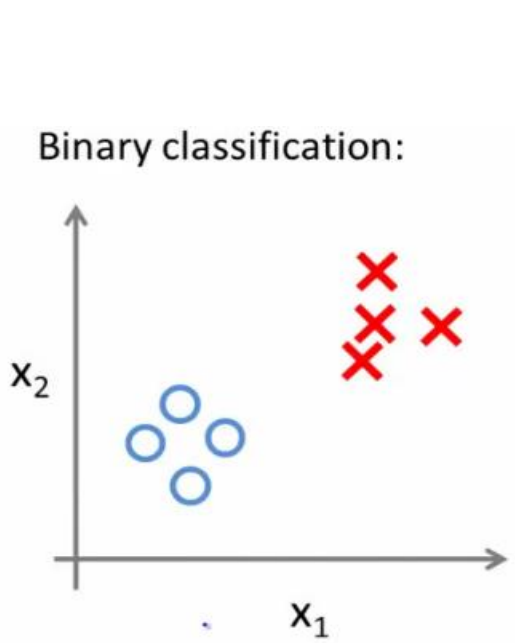
Credits: Some slides adapted from Fayyaz Minhas

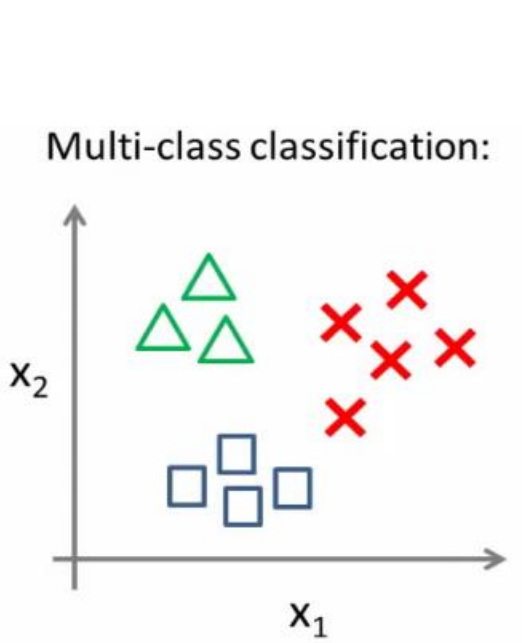
Evaluation of Multiclass Classification

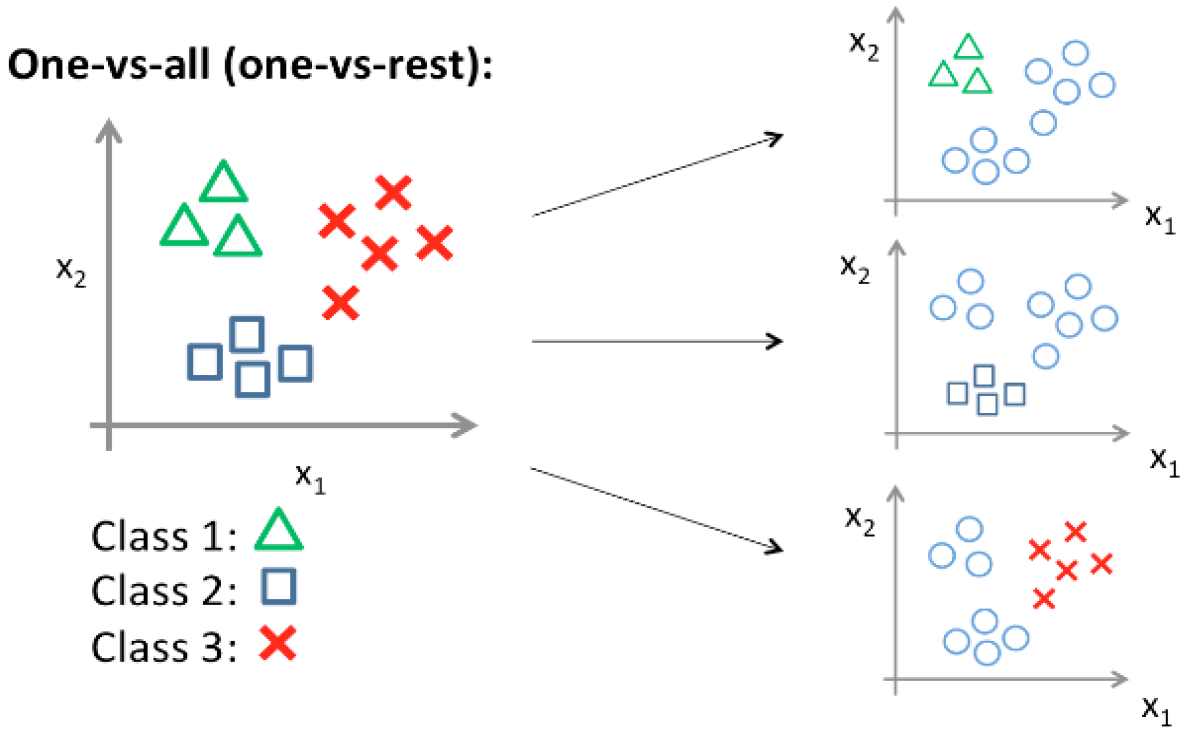
3. Choosing a Classifier

- How many categories (k)?
  - [k=2] **Binary** → binary classifier
  - [k>2] **Multiclass**:
    - One-vs-all classifiers
      Build *k* classifiers, each able to distinguish class *i* from the rest. Then combine output of all classifiers (e.g. based on their confidence scores)
    - Multinomial/multiclass classifiers

Binary vs. Multiclass Classifier







When to use Multinomial or One-vs-All?

- **Multinomial**:
  - is generally faster, a single classifier.
  - classes are mutually exclusive, no overlap.
- **One-vs-all**:
  - **Multilabel classification**, document can fall in **1+ categories**:
    e.g. classify by language:

    I said “bonjour mon ami” to my friend → English & French

    How? Out of *k* classifiers, those with **confidence > threshold**

Evaluation of Multiclass Classification

- Bigger **confusion matrix**:

	Actually UK	Actually World	Actually Tech	Actually Science	Actually Politics	Actually Business
Classified as UK	**95**	1	13	0	1	0
Classified as World	0	**1**	0	0	0	0
Classified as Tech	10	90	**0**	1	0	0
Classified as Science	0	0	0	**34**	3	7
Classified as Politics	0	1	2	13	**26**	5
Classified as Business	0	0	2	14	5	**10**

Overall Accuracy

- **Overall Accuracy:** ratio of correct classifications.

$$\frac{\sum_{i} c_{ii}}{\sum_{j}\sum_{i} c_{ij}}$$

Only when the dataset is **balance**!

- **Generally a bad evaluation approach**, e.g.:
  - Dataset with 1,000 emails: 970 are legitimate, 20 phishing, 10 spam.
  - Our classifier predicts all emails as legitimate.
  - 970 classified correctly: 970 / 1000 = 0.97 accuracy!

Misleading

Per-Class Precision / Recall

- **Per-class precision and recall**:

Precision: $\frac{c_{ii}}{\sum_j c_{ji}}$

→ # of items correctly classified as class i

→ # of items classified as class i

Recall: $\frac{c_{ii}}{\sum_j c_{ij}}$

→ # of items correctly classified as class i

→ # of actual class i items

- With the **harmonic mean**, we can then get **per-class F1 score**.

Overall Performance

- We have **per-class precision, recall** and **F1 scores**.
  - How do we combine them all to **get a single score**?

Overall Performance

Obtaining overall performances:

- **Macro-averaging:**
  Compute **performance independently for each class, then average them**.
  → All **classes are given equal weights** to the final score, regardless of their size (e.g. class with 990 and class with 10 instances).

- **Micro-averaging:**
  Compute **overall performance** without computing per-class performances.
  → **Large classes contribute more** to the final score.

Macro-Averaging

- Macro-averaging:

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}}{k}$$

$$\text{macroaveraged recall} = \frac{\sum_i \frac{\text{TP}}{\text{TP}_i + \text{FN}_i}}{k}$$

- The macro-averaged F1 score is then the harmonic mean of those.

Micro-Averaging

- Micro-averaging:

$$\text{microaveraged precision} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \text{FP}_i}$$

$$\text{microaveraged recall} = \frac{\sum_i \text{TP}}{\sum_i \text{TP}_i + \text{FN}_i}$$

- The micro-averaged F1 score is then the harmonic mean of those.

Micro- vs. Macro-Averaging Example

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

- Micro-averaged F1 score: **0.66**
- Macro-averaged F1 score: **0.45**

Q: where I have so many instances with label C, which metric is fairer?

A: depends on our objective: do we need a classifier that performs well for all labels? Do we just want to perform well for C?

Micro- vs. Macro-Averaging Example

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Micro- vs. Macro-Averaging Example

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Micro- vs. Macro-Averaging Example

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$$\text{Precision} = 366/907 = 0.40$$

Micro- vs. Macro-Averaging Example

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$$\text{Precision} = 366/907 = 0.40$$

$$\text{Recall} = 366/676 = 0.54$$

Macro-averaged F1 calculation

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$$\text{Precision} = 366/907 = 0.40$$
$$\text{Recall} = 366/676 = 0.54$$

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

$$\text{Precision} = 22/343 = 0.06$$
$$\text{Recall} = 22/155 = 0.14$$

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

$$\text{Precision} = 149/358 = 0.42$$
$$\text{Recall} = 149/327 = 0.46$$

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

$$\text{Precision} = 2421/2906 = 0.83$$
$$\text{Recall} = 2421/3356 = 0.72$$

Macro-averaged F1 calculation

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}}{k}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$\text{Precision} = 366/907 = 0.40$
$\text{Recall} = 366/676 = 0.54$

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

$\text{Precision} = 22/343 = 0.06$
$\text{Recall} = 22/155 = 0.14$

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

$\text{Precision} = 149/358 = 0.42$
$\text{Recall} = 149/327 = 0.46$

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

$\text{Precision} = 2421/2906 = 0.83$
$\text{Recall} = 2421/3356 = 0.72$

$$\text{MacroAvg} - \text{Precision} = \frac{0.40 + 0.06 + 0.42 + 0.83}{4} = 0.43$$

Macro-averaged F1 calculation

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}}{k}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$$\text{Precision} = 366/907 = 0.40$$
$$\text{Recall} = 366/676 = 0.54$$

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

$$\text{Precision} = 22/343 = 0.06$$
$$\text{Recall} = 22/155 = 0.14$$

$$\text{MacroAvg} - \text{Precision}$$
$$= \frac{0.40 + 0.06 + 0.42 + 0.83}{4}$$
$$= 0.43$$

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

$$\text{Precision} = 149/358 = 0.42$$
$$\text{Recall} = 149/327 = 0.46$$

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

$$\text{Precision} = 2421/2906 = 0.83$$
$$\text{Recall} = 2421/3356 = 0.72$$

Macro-averaged F1 calculation

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}}{k}$$

$$\text{macroaveraged recall} = \frac{\sum_i \frac{\text{TP}}{\text{TP}_i+\text{FN}_i}}{k}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Precision = 366/907 = 0.40
Recall = 366/676 = 0.54

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

Precision = 22/343 = 0.06
Recall = 22/155 = 0.14

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

Precision = 149/358 = 0.42
Recall = 149/327 = 0.46

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

Precision = 2421/2906 = 0.83
Recall = 2421/3356 = 0.72

MacroAvg – Precision

$$= \frac{0.40 + 0.06 + 0.42 + 0.83}{4}$$

$$= 0.43$$

MacroAvg – Recall

$$= \frac{0.54 + 0.14 + 0.46 + 0.72}{4}$$

$$= 0.43$$

Macro-averaged F1 calculation

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}}{k}$$

$$\text{macroaveraged recall} = \frac{\sum_i \frac{\text{TP}}{\text{TP}_i+\text{FN}_i}}{k}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Precision = 366/907 = 0.40
Recall = 366/676 = 0.54

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

Precision = 22/343 = 0.06
Recall = 22/155 = 0.14

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

Precision = 149/358 = 0.42
Recall = 149/327 = 0.46

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

Precision = 2421/2906 = 0.83
Recall = 2421/3356 = 0.72

MacroAvg – Precision

$$= \frac{0.40 + 0.06 + 0.42 + 0.83}{4}$$

$$= 0.43$$

MacroAvg – Recall

$$= \frac{0.54 + 0.14 + 0.46 + 0.72}{4}$$

$$= 0.47$$

MacroAvg – F1

$$= \frac{2 \times 0.43 \times 0.47}{0.43 + 0.47}$$

$$= 0.45$$

Macro-averaged F1 calculation

$$\text{macroaveraged precision} = \frac{\sum_i \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}}{k}$$

$$\text{macroaveraged recall} = \frac{\sum_i \frac{\text{TP}}{\text{TP}_i+\text{FN}_i}}{k}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Precision = 366/907 = 0.40
Recall = 366/676 = 0.54

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

Precision = 22/343 = 0.06
Recall = 22/155 = 0.14

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

Precision = 149/358 = 0.42
Recall = 149/327 = 0.46

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

Precision = 2421/2906 = 0.83
Recall = 2421/3356 = 0.72

MacroAvg – Precision

$$= \frac{0.40 + 0.06 + 0.42 + 0.83}{4}$$
$$= 0.43$$

MacroAvg – Recall

$$= \frac{0.54 + 0.14 + 0.46 + 0.72}{4}$$
$$= 0.47$$

MacroAvg – F1

$$= \frac{2 \times 0.43 \times 0.47}{0.43 + 0.47}$$
$$= 0.45$$

Micro-averaged F1 calculation

$$\text{microaveraged precision} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \text{FP}_i}$$

$$\text{microaveraged recall} = \frac{\sum_i \text{TP}}{\sum_i \text{TP}_i + \text{FN}_i}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

Precision = 366/907 = 0.40
Recall = 366/676 = 0.54

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

Precision = 22/343 = 0.06
Recall = 22/155 = 0.14

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

Precision = 149/358 = 0.42
Recall = 149/327 = 0.46

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

Precision = 2421/2906 = 0.83
Recall = 2421/3356 = 0.72

MicroAvg – Precision

$$= \frac{366 + 22 + 149 + 2421}{907 + 343 + 358 + 2906} = 0.66$$

MicroAvg – Recall

$$= \frac{366 + 22 + 149 + 2421}{676 + 155 + 327 + 3356} = 0.66$$

MicroAvg – F1

$$= \frac{2 \times 0.66 \times 0.66}{0.66 + 0.66} = 0.66$$

Micro-averaged F1 calculation

$$\text{microaveraged precision} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \text{FP}_i}$$

$$\text{microaveraged recall} = \frac{\sum_i \text{TP}}{\sum_i \text{TP}_i + \text{FN}_i}$$

	S	Not S	Total
S	366	541	**907**
Not S	310	3297	**3607**
Total	**676**	**3838**	

$\text{Precision} = 366/907 = 0.40$
$\text{Recall} = 366/676 = 0.54$

	D	Not D	Total
D	22	321	**343**
Not D	133	4038	**4171**
Total	**155**	**4359**	

$\text{Precision} = 22/343 = 0.06$
$\text{Recall} = 22/155 = 0.14$

	Q	Not Q	Total
Q	149	209	**358**
Not Q	178	3978	**4156**
Total	**327**	**4187**	

$\text{Precision} = 149/358 = 0.42$
$\text{Recall} = 149/327 = 0.46$

	C	Not C	Total
C	2421	485	**2906**
Not C	935	673	**1608**
Total	**3356**	**1158**	

$\text{Precision} = 2421/2906 = 0.83$
$\text{Recall} = 2421/3356 = 0.72$

$$\text{MicroAvg} - \text{Precision} = \frac{366 + 22 + 149 + 2421}{907 + 343 + 358 + 2906} = 0.66$$

$$\text{MicroAvg} - \text{Recall} = \frac{366 + 22 + 149 + 2421}{676 + 155 + 327 + 3356} = 0.66$$

$$\text{MicroAvg} - \text{F1} = \frac{2 \times 0.66 \times 0.66}{0.66 + 0.66} = 0.66$$

Further Weighting / Selection

- We can choose to **prioritise certain categories**:
  - Give higher weight to important categories:
    0.3*Prec(c_1) + **0.3***Prec(c_2) + **0.4***Prec(c_3)
    → unless we have clear criteria to choose these weights, it's rather arbitrary though!
  - Select some categories for inclusion in macro/microaverage:
    - We can decide to only macroaverage over positive and negative classes. Performance over the neutral class will not included.

ROC: From two classes to three

A symptom-checker diagnoses **influenza**, **COVID-19** or a **common cold**.

Each patient has **one true class** but receives **three scores**: the estimated probability of each disease. The prediction is the highest one.

With three classes, which patients count as positive, and which as negative?

#	True class	P(flu)	P(COVID)	P(cold)	Predicted
1	Influenza	**80%**	15%	5%	Influenza
2	Influenza	**60%**	10%	30%	Influenza
3	COVID-19	20%	**65%**	15%	COVID-19
4	COVID-19	15%	**45%**	40%	COVID-19
5	Common cold	10%	35%	**55%**	Common cold
6	Common cold	25%	**50%**	25%	COVID-19

ROC: From two classes to three

A symptom-checker diagnoses **influenza**, **COVID-19** or a **common cold**.

Each patient has **one true class** but receives **three scores**: the estimated probability of each disease. The prediction is the highest one.

With three classes, which patients count as positive, and which as negative?

#	True class	P(flu)	P(COVID)	P(cold)	Predicted
1	Influenza	**80%**	15%	5%	Influenza
2	Influenza	**60%**	10%	30%	Influenza
3	COVID-19	20%	**65%**	15%	COVID-19
4	COVID-19	15%	**45%**	40%	COVID-19
5	Common cold	10%	35%	**55%**	Common cold
6	Common cold	25%	**50%**	25%	COVID-19

Multiclass confusion matrix

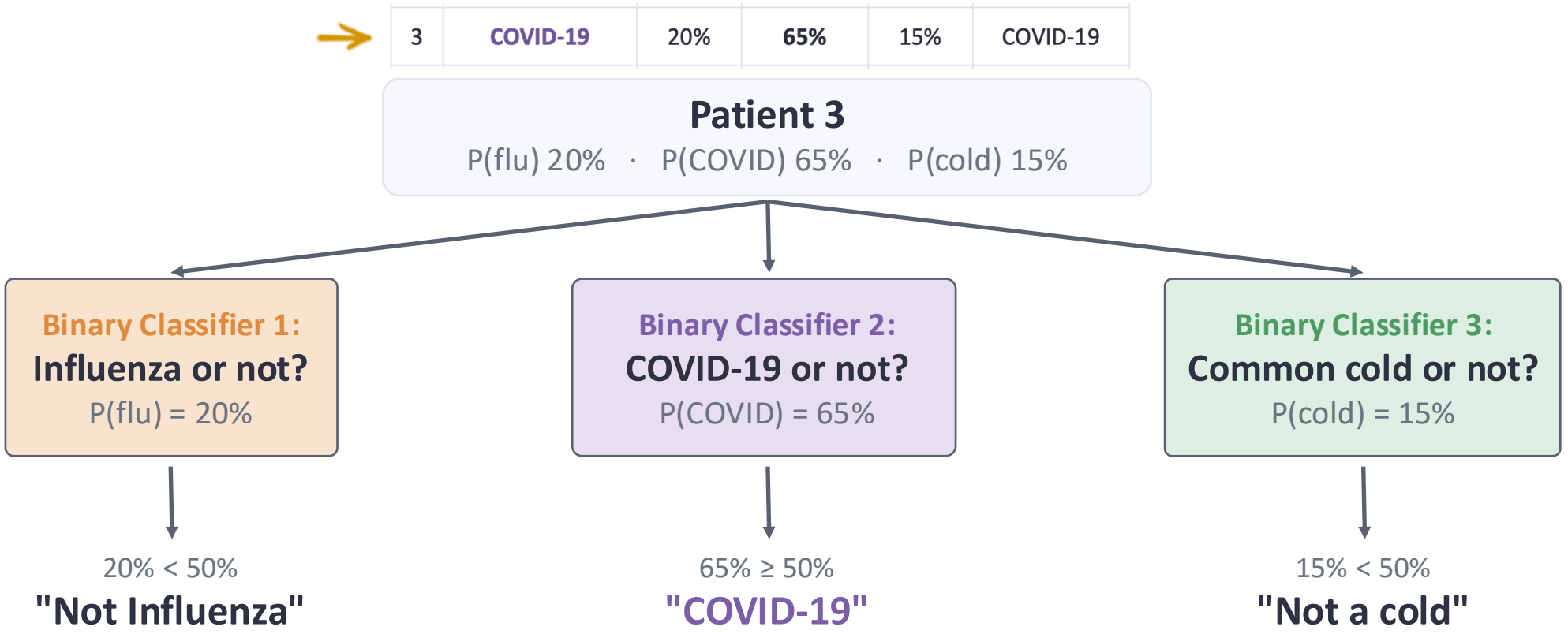
#	True class	P(flu)	P(COVID)	P(cold)	Predicted
1	Influenza	**80%**	15%	5%	Influenza
2	Influenza	**60%**	10%	30%	Influenza
3	COVID-19	20%	**65%**	15%	COVID-19
4	COVID-19	15%	**45%**	40%	COVID-19
5	Common cold	10%	35%	**55%**	Common cold
6	Common cold	25%	**50%**	25%	COVID-19

Predicted \ Actual	Influenza	COVID-19	Common cold
Influenza	2	0	0
COVID-19	0	2	1
Common cold	0	0	1

Accuracy = 5 / 6 ≈ 83%

- We look at the possible scores and build binary problems.

Binary classification: **One vs rest**



- One classifier supplies all three scores; one-vs-rest reads each score as the answer to its own yes/no question.
- **Sliding** one question's **threshold** gives its ROC curve.

COVID vs rest: one threshold, one matrix

#	Has COVID?	P(COVID)	Predicted at ≥ 50%
1	No	15%	Not COVID (TN)
2	No	10%	Not COVID (TN)
3	Yes	**65%**	COVID (TP)
4	Yes	45%	Not COVID (FN)
5	No	35%	Not COVID (TN)
6	No	**50%**	COVID (FP)

		Actual: COVID	Actual: Not COVID
Predicted	COVID	1	1
Predicted	Not COVID	1	3

TPR = 1/2 = 0.5
FPR = 1/4 = 0.25
One ROC point (0.25, 0.5)

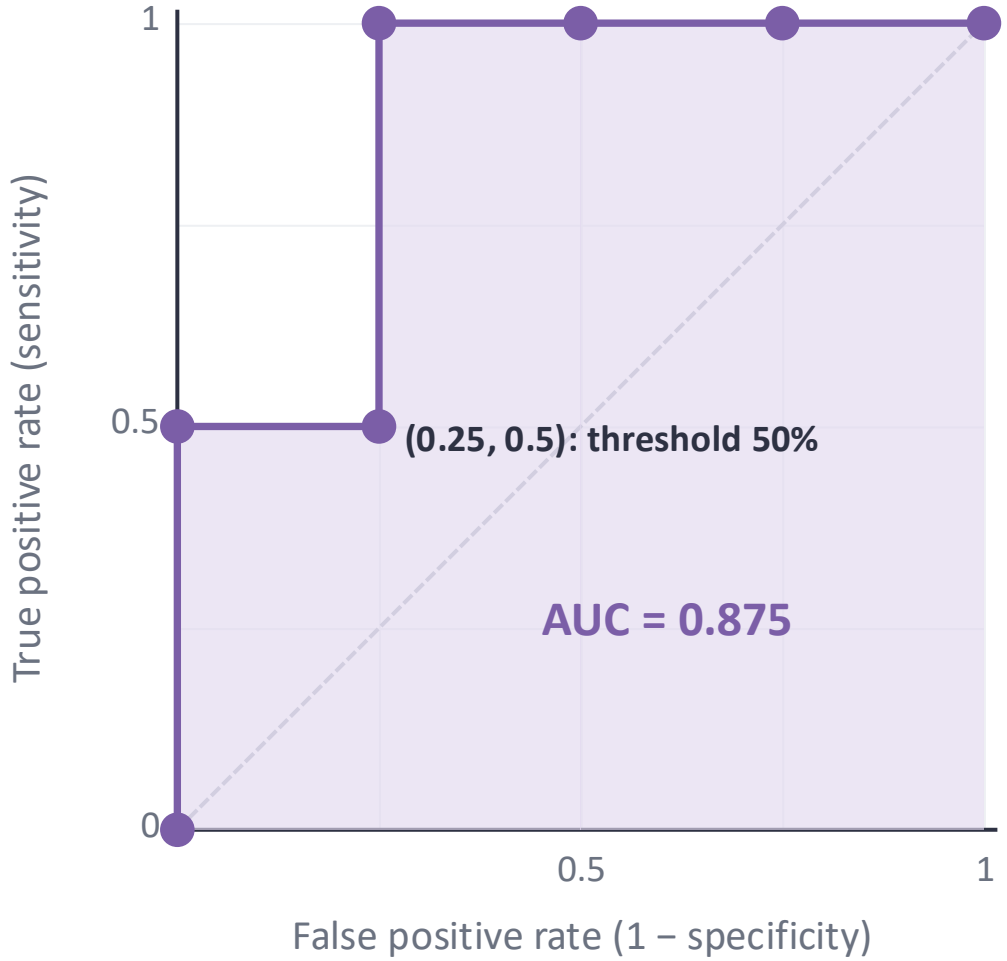
Subtlety: patient 4 is correctly predicted as COVID in the multiclass matrix (45% is their highest score), but is **not COVID** at a 50% threshold. Argmax decisions and one-vs-rest thresholds are not the same.

Sliding the threshold traces the ROC curve

Threshold (≥)	TP / FP	TPR	FPR
> 65%	0 / 0	0.00	0.00
65%	1 / 0	0.50	0.00
50%	1 / 1	0.50	0.25
45%	2 / 1	1.00	0.25
35%	2 / 2	1.00	0.50
15%	2 / 3	1.00	0.75
10%	2 / 4	1.00	1.00

Each threshold gives a confusion matrix and one point.

Area under the curve = 0.25 × 0.5 + 0.75 × 1 = 0.875



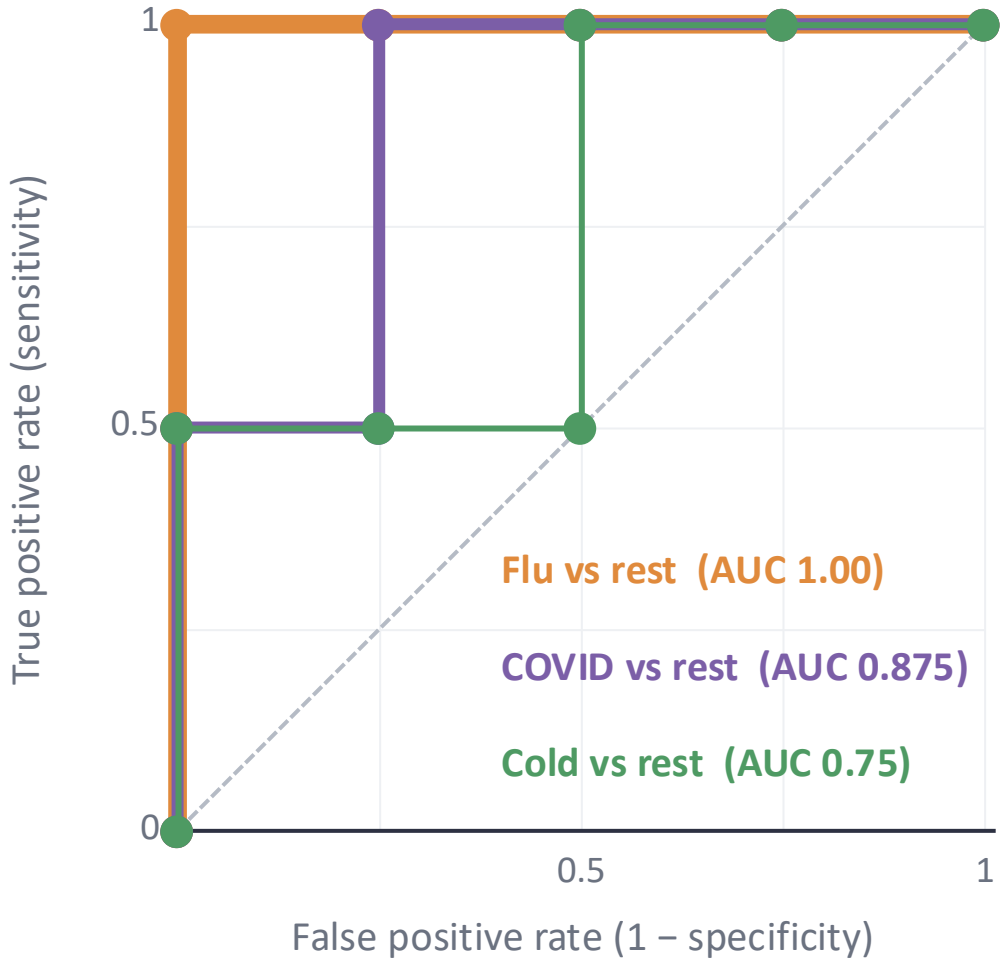
One vs rest: Three classes, three ROC curves

Computed from the six patients, one probability column at a time.

Influenza is perfectly separated; the common cold is the hardest class.

Important: each curve ranks one class's score against the others; it does not directly measure the final decision, which picks the highest score.

How to combine the different AUC scores?



One number: **the macro-average AUC**

$$\textbf{Macro-AUC} = (1/K) \times \Sigma_k \text{AUC}_k$$

Class	One-vs-rest AUC
Influenza	1.00
COVID-19	0.875
Common cold	0.75

Average the per-class AUCs, giving every class the same weight:

For our six patients

(1.00 + 0.875 + 0.75) / 3 = 0.875

- Every class counts equally, however many patients have it.

On the full test set: **macro versus weighted**

$$\text{Macro-AUC} = (1/K) \times \Sigma_k \, AUC_k$$

$$\textbf{Weighted-AUC} = \Sigma_k \, (\mathbf{n_k} / N) \times AUC_k$$

Now the full test set: 1,000 patients, 800 of whom have a cold.

Class	ROC-AUC	Patients	Share
Influenza	0.95	100	10%
COVID-19	0.85	100	10%
Common cold	0.70	800	80%

Macro

(0.95 + 0.85 + 0.70) / 3 = 0.833

Every class counts equally.

Weighted

0.1 × 0.95 + **0.1** × 0.85 + **0.8** × 0.70 = 0.74

The common cold, 80% of patients
and the hardest class, pulls it down.

Micro-average AUC: pool everything

Pool all one-vs-rest decisions and draw a **single** ROC curve.

Each patient gives one (score, label) pair per class.

6 patients × 3 classes = 18 pairs;

e.g.: **micro-AUC = 0.924.**

Frequent classes contribute more pairs, so weak performance on a rare class can be hidden.

Patient	Class	Score	Has this disease?
1	Flu	80%	Yes (positive)
	COVID	15%	No (negative)
	Cold	5%	No (negative)
3	Flu	20%	No (negative)
	COVID	65%	Yes (positive)
	Cold	15%	No (negative)

…and so on for all six patients. All **18 pairs** go into one ROC curve.

One-vs-one: every pair, in both directions

- Only the patients of the two classes are used. **Matrices at a 50% threshold**; **AUC over all thresholds**.
- The two directions may be different because of the probability taken by the third class.

Flu vs **COVID**
positive: Flu
rank by P(flu)

	Flu	COVID
Flu	2	0
COVID	0	2

AUC
1.00

COVID vs **Flu**
positive: COVID
rank by P(COVID)

	COVID	Flu
COVID	1	0
Flu	1	2

AUC
1.00

Flu vs **Cold**
positive: Flu
rank by P(flu)

	Flu	Cold
Flu	2	0
Cold	0	2

AUC
1.00

Cold vs **Flu**
positive: Cold
rank by P(cold)

	Cold	Flu
Cold	1	0
Flu	1	2

AUC
0.75

COVID vs **Cold**
positive: COVID
rank by P(COVID)

	COVID	Cold
COVID	1	1
Cold	1	1

AUC
0.75

Cold vs **COVID**
positive: Cold
rank by P(cold)

	Cold	COVID
Cold	1	0
COVID	1	2

AUC
0.75

3 pairs × 2 directions = 6 binary problems. Pair AUC = average of its two directions.

Why do flu vs cold and cold vs flu differ?

#	True class	P(flu)	P(cold)	P(COVID)
1	Influenza	**80%**	5%	15%
2	Influenza	**60%**	**30%**	10%
5	Common cold	10%	**55%**	35%
6	Common cold	25%	**25%**	**50%**

Flu vs cold: read P(flu)

Does the model recognise flu?

80%, 60% above 10%, 25%: 4 of 4 **AUC: 1.00**

Cold vs flu: read P(cold)

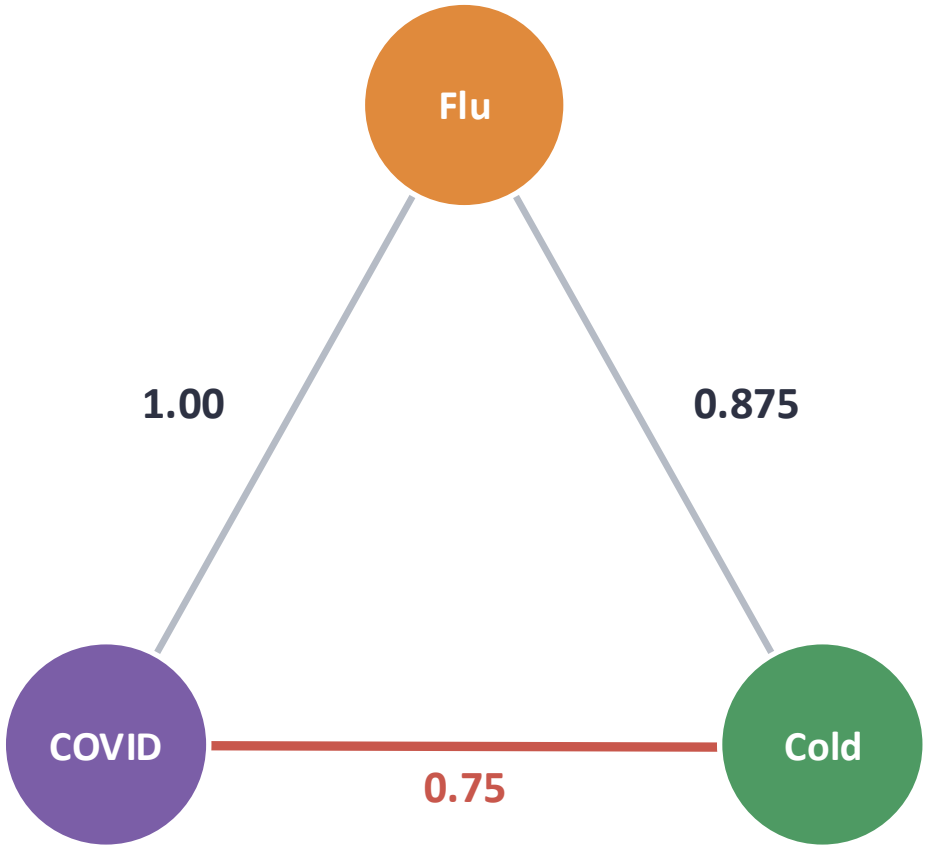
Does the model recognise a cold?

Patient 6's 25% < patient 2's 30%: 3 of 4 **AUC: 0.75**

- **Why the columns are not mirror images:** with only two classes, P(cold) = 100% − P(flu) and both directions agree. With three, part of the probability goes to COVID-19, in different amounts for each patient (50% for patient 6, 10% for patient 2).
- Neither direction is preferred: we average them. A large gap shows which class's score is weak, here the cold.

One-vs-one: repeat for every pair, then average

Pair	Rank by first class	Rank by second class	Pair AUC
Flu vs **COVID**	1.00 (P(flu))	1.00 (P(COVID))	**1.00**
Flu vs **Cold**	1.00 (P(flu))	0.75 (P(cold))	**0.875**
COVID vs **Cold**	0.75 (P(COVID))	0.75 (P(cold))	**0.75**
Mean over pairs			**0.875**



- **What one-vs-one adds:** it shows which pair of diseases is hard to tell apart.
- COVID-19 and the common cold, the confusion we saw with patient 6.

- Number of pairs = K(K − 1) / 2:
  3 classes → 3 pairs, 10 classes → 45 pairs.

Which approach should you use?

Approach	Main advantage	Limitation
One-vs-rest, macro	Every class has equal importance	Each negative group mixes potentially very different classes
One-vs-rest, weighted	Reflects how frequent each class is	Majority classes can dominate
One-vs-rest, micro	A single pooled ROC curve	Can hide poor performance on individual classes
One-vs-one, macro	Evaluates how well each pair of classes is separated	The number of comparisons grows quadratically

Good default: one-vs-rest with the macro average, the simplest extension of the binary ROC curve, reported together with the per-class AUCs.

In Python with scikit-learn

```
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt

# y_true: true labels (0, 1, 2); y_prob: class probabilities, shape (n, 3)
names = ["Influenza", "COVID-19", "Common cold"]
for k, name in enumerate(names):           # one ROC curve per class
    fpr, tpr, _ = roc_curve(y_true == k, y_prob[:, k])
    plt.plot(fpr, tpr, label=f"{name} vs rest")
plt.plot([0, 1], [0, 1], "--", color="grey")
plt.xlabel("False positive rate"); plt.ylabel("True positive rate")
plt.legend(); plt.show()

macro    = roc_auc_score(y_true, y_prob, multi_class="ovr", average="macro")
weighted = roc_auc_score(y_true, y_prob, multi_class="ovr", average="weighted")
micro    = roc_auc_score(label_binarize(y_true, classes=[0, 1, 2]), y_prob,
average="micro")
ovo      = roc_auc_score(y_true, y_prob, multi_class="ovo", average="macro")
```

The key idea: decide what to compare

ROC-AUC measures **ranking**: how often positive examples score higher than negative ones.

With two classes, it is obvious what to compare. With many classes, we must first decide: one class against the rest (one-vs-rest), or one class against another (one-vs-one).

With imbalanced classes, report together:

Per-class AUCs	Show which classes the model handles well, and which it does not.
Macro-AUC	One summary number in which every class counts equally.
Multiclass confusion matrix	Shows which classes are confused with which, at the actual decisions.

Error Analysis

Error Analysis

- **Error analysis:** can help us find out where our classifier can do better.
- No magic formula for performing error analysis.
  - Look where we are doing wrong, what labels particularly.
  - Do our errors have some common characteristics? Can we infer a new feature from that?
  - Could our classifier be favouring one of the classes (e.g. the majority class)?

Error Analysis

- **Error analysis:** where are we doing wrong? What labels?

  Look at frequent deviations in the confusion matrix.

	Actually UK	Actually World	Actually Tech	Actually Science	Actually Politics	Actually Business
Classified as UK	**95**	1	13	0	1	0
Classified as World	0	**1**	0	0	0	0
Classified as Tech	10	90	**0**	1	0	0
Classified as Science	0	0	0	**34**	3	7
Classified as Politics	0	1	2	13	**26**	5
Classified as Business	0	0	2	14	5	**10**

Error Analysis

Error analysis: could our classifier be favouring one of the classes?

- Owing to class imbalance, classifiers tend to predict ***popular classes*** more often.
  E.g.: class A (700), class B (100), class C (100), class D (100)
  classifiers will tend to predict A.

Over-represented as in our previous example:

	S	D	Q	C
S	**366 (40.4%)**	32 (3.5%)	22 (2.4%)	487 (53.7%)
D	38 (11.1%)	**22 (6.4%)**	23 (6.7%)	260 (75.8%)
Q	11 (3.1%)	10 (2.8%)	**149 (41.6%)**	188 (52.5%)
C	261 (9.0%)	91 (3.1%)	133 (4.6%)	**2,421 (83.3%)**

How to Deal with Class Imbalance?

How to deal with imbalance (e.g. **A-700, B-100, C-100, D-100**)?

- **Undersample popular class** → **A-100**, B-100, C-100, D-100

  randomly remove 600 instances of A

- **Oversample other classes** → A-700, **B-700, C-700, D-700**

  repeat instances of B, C, D to match the number of A's

- **Create synthetic data** → A-700, **B-700, C-700, D-700**

  generate new B, C, D items.

  Needs understanding of the contents of the classes to produce sensible data items

- **Cost sensitive learning**

  e.g. higher probability to predict uncommon classes

  P(A)=1/700, P(B)=1/100, P(C)=1/100, P(D)=1/100

  `scikit → class_weight="auto"`

Reading

James, Witten, Hastie, Tibshirani and Taylor (2023). "***An Introduction to Statistical Learning with Applications in Python (ISLP)***".

- 4.4.2 Confusion matrix, sensitivity, specificity and ROC curves (revision)
- 4.3.5 Multinomial logistic regression: classifiers that output one probability per class
- 9.4 SVMs with more than two classes: one-versus-one and one-versus-all
- Lab 9.6.3 Drawing ROC curves in Python with scikit-learn

Hastie, Tibshirani and Friedman (2009). "***The Elements of Statistical Learning (ESL), 2nd edition.***"

- 9.2.5 ROC curves and AUC (revision)

Hand and Till (2001), "***A simple generalisation of the area under the ROC curve for multiple class classification problems"***, Machine Learning 45, 171–186