School of Electronic Engineering and Computer Science
Queen Mary University of London

ECS7020P Principles of Machine Learning
Week 5: Classification II

Dr Jesús Requena Carrión

26 Oct 2021

The best diagnostic machine

As the hospital lead data scientist, you are responsible for selecting the best diagnostic machine for a certain disease. You are presented three machines A, B and C, which **you test** using a group of patients whose diagnose you already know (**labelled dataset**).

The resulting accuracy of each machine after testing is shown below. Which one would you choose?

(a) Machine A: 6 % of correct diagnoses

(b) Machine B: 89 % of correct diagnoses

(c) Machine C: 56 % of correct diagnoses

What is the most appropriate quality metric?

Despite previous slip-ups, you are still the hospital's lead data scientist and ask yourself what the most useful metric for evaluating a diagnostic machine could be. Which option would you choose among the following?

(a) Accuracy, i.e. proportion of correct diagnoses

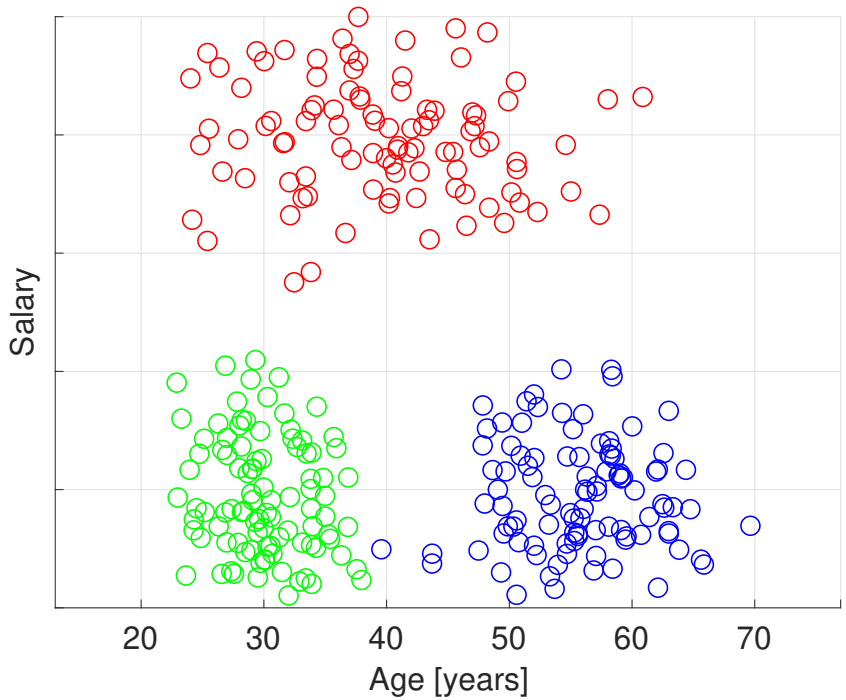
(b) Either proportion of healthy patients correctly identified or proportion of ill patients correctly identified

(c) Something else

Know your metrics!

Classifiers in the predictor space

When labels are discrete, datasets can be visualised as **collections of points in the predictor space**, where symbols represent a different label.



Opinion: o → Good, o → Neutral, o → Bad

- A classifier is a partition of the space into **decision regions**.
- We use a **dataset** together with a notion of **quality** to find the best partition.

The logistic model and kNN

We have introduced the **accuracy** and **error rate** as two convenient metrics to measure the quality of a classifier.

The following two approaches to build a classifier have been discussed:

- **Logistic model**: Trains a linear model using the likelihood or log-likelihood function as a quality metric during optimisation.
- **kNN**: Instance-based model that simply compares the proportion of samples of each class within a neighbourhood.

Notice that neither of them seem to be defined based on the notions of accuracy and error rate. Do these notions play any role at all?

Agenda

The Bayes classifier

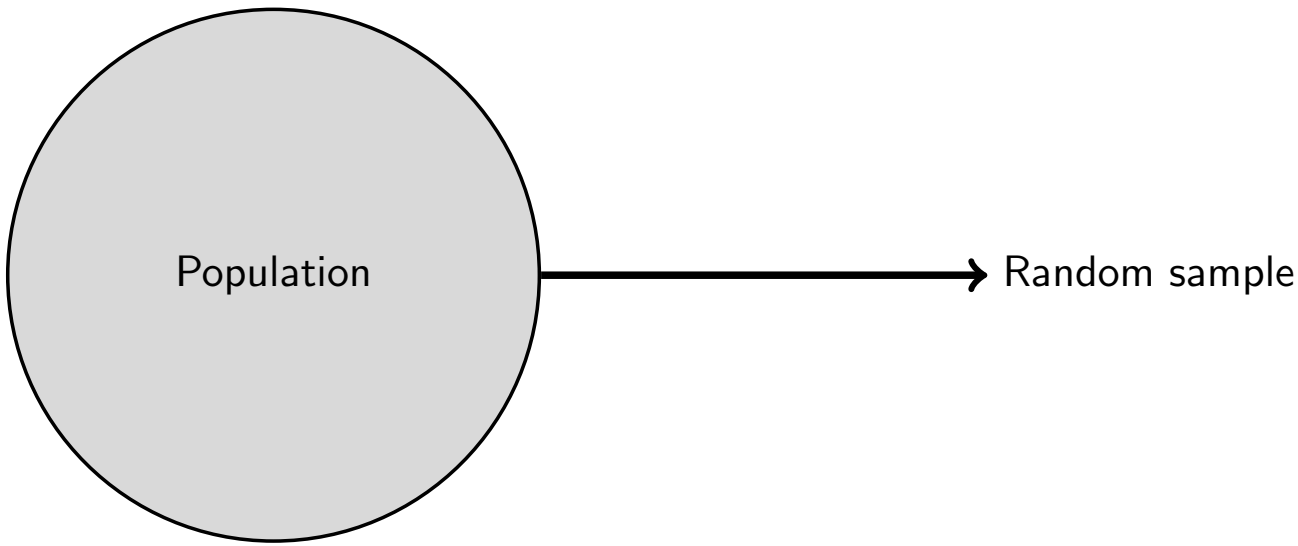
Using data to build posterior probabilities

Classification performance: Beyond accuracy

Summary

Which classifier has the highest accuracy?

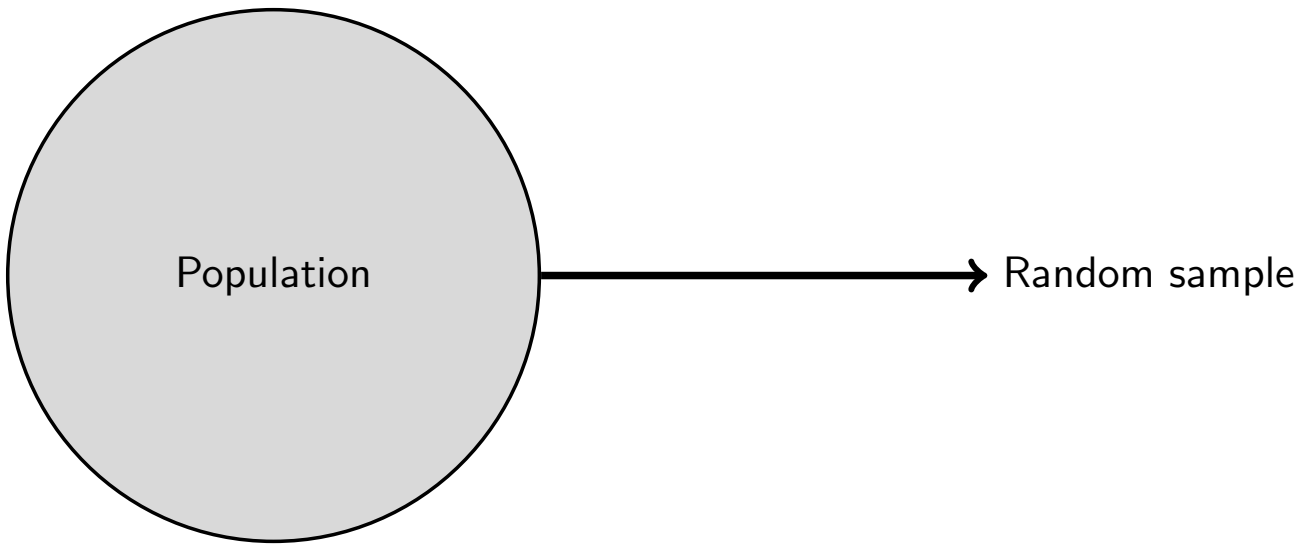
Consider a population consisting of individuals with an attribute y that can take on two values: o or o.



How do we determine whether a sample extracted randomly belongs to class o or o? How can we use any **available information** to achieve the **highest classification accuracy**?

The coin as a classifier

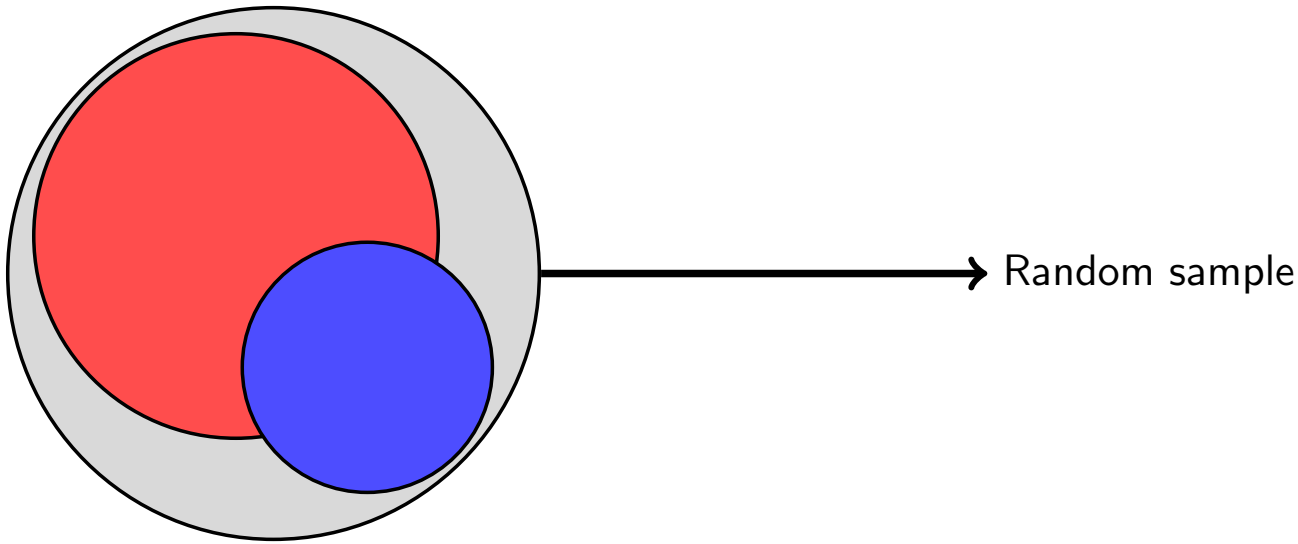
Without any additional information, we have no basis to decide which class the sample belongs to.



Flipping a o / o coin would give us the best guess. *Fifty-fifty*.

Prior probabilities

Now assume that we know that in 70 % of the population y = o, whereas in the remaining 30 % y = o.



In this scenario, we know the **class priors**, namely $P(y = \text{o}) = 0.7$ and $P(y = \text{o}) = 0.3$. We would **always** choose o.

Class densities

What if we have access to the value of **another attribute** x and know **how frequently** a value of x appears within each class?

For instance, assume we have the following insight:

- In $1/4$ of the o samples, $x = a$ and in $3/4$ $x = b$.
- In $2/3$ of the o samples $x = a$ and in $1/3$ $x = b$.

These are **class densities** and are expressed as follows:

$$p(x = a|y = \text{o}) = 1/4, \qquad p(x = b|y = \text{o}) = 3/4$$
$$p(x = a|y = \text{o}) = 2/3, \qquad p(x = b|y = \text{o}) = 1/3$$

Given a sample where $x = a$, we could compare $p(x = a|y = \text{o})$ and $p(x = a|y = \text{o})$ and decide that the sample belongs to o.

Posterior probabilities and the Bayes classifier

It turns out that the classifier that achieves the **highest accuracy** is the one that compares the **posterior probabilities**, defined as the probability that a sample belongs to a class given all the available information.

If our information about a sample is $x = a$, the posterior probabilities are:

$$P(y = \text{o} \,|x = a)$$
$$P(y = \text{o} \,|x = a)$$

Do not confuse a posterior probability with the a class density! This confusion is called the **fallacy of the transposed conditional**.

The classifier that uses the **true posterior probabilities** is called the **Bayes classifier**. The question is, *how do we get the posterior probabilities*?

Agenda

Machine learning as statistical learning

From a statistical point of view, machine learning classifiers use **datasets** to estimate **posterior probabilities**.

For instance:

- **Logistic models** use the logistic function to build a posterior probability (the classifier's *certainty*).
- In **kNN** the posterior probabilities are estimated as the proportion of neighbours that belong to each class .

These methods provide **approximated posterior probabilities** and will never beat a Bayes classifier, which uses the **true posterior probability**.

Bayes rule

If we happen to know the **priors** and the **class densities**, we can apply Bayes rule to obtain the **posterior probabilities** exactly:

$$P(y = \text{o} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = \text{o})P(y = \text{o})}{p(\boldsymbol{x})}, \quad P(y = \text{o} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = \text{o})P(y = \text{o})}{p(\boldsymbol{x})}$$

The Bayes classifier uses the **odds ratio** to classify a sample:

$$\frac{P(y = \text{o} \mid \boldsymbol{x})}{P(y = \text{o}|\boldsymbol{x})} = \frac{p(\boldsymbol{x}|y = \text{o})P(y = \text{o})}{p(\boldsymbol{x}|y = \text{o})P(y = \text{o})} \lessgtr 1$$

If the ratio is greater than 1, the sample is o, if it is less, it is o. This is equivalent to assigning the sample to the most probable class.

Bayes rule in machine learning

In machine learning, we can use **data** to estimate the **priors** and the **class densities**.

Using a dataset to estimate the **priors** is very easy, we simply need to count the number of samples belonging to each class:

$$P(y = \text{o}) = \frac{\#\ \text{o samples}}{\#\ \text{samples}}, \quad P(y = \text{o}) = \frac{\#\ \text{o samples}}{\#\ \text{samples}}$$

Building the **class densities** $p(\boldsymbol{x}|y = \text{o})$ and $p(\boldsymbol{x}|y = \text{o})$ is in fact an **unsupervised problem** (more about this, in week 8).

For now, we will focus on the most common density, the **Gaussian density**, on which **discriminant analysis** techniques are based.

Discriminant analysis

In **discriminant analysis**, we assume that the class densities are **Gaussian**. If there is one predictor x, the o class density is:

$$p(x|y = \text{o}) = \frac{1}{\sigma_\text{o}\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_\text{o}}{\sigma_\text{o}})^2}$$

where μ_o is the **mean** and σ^2_o the **variance** of the Gaussian density.

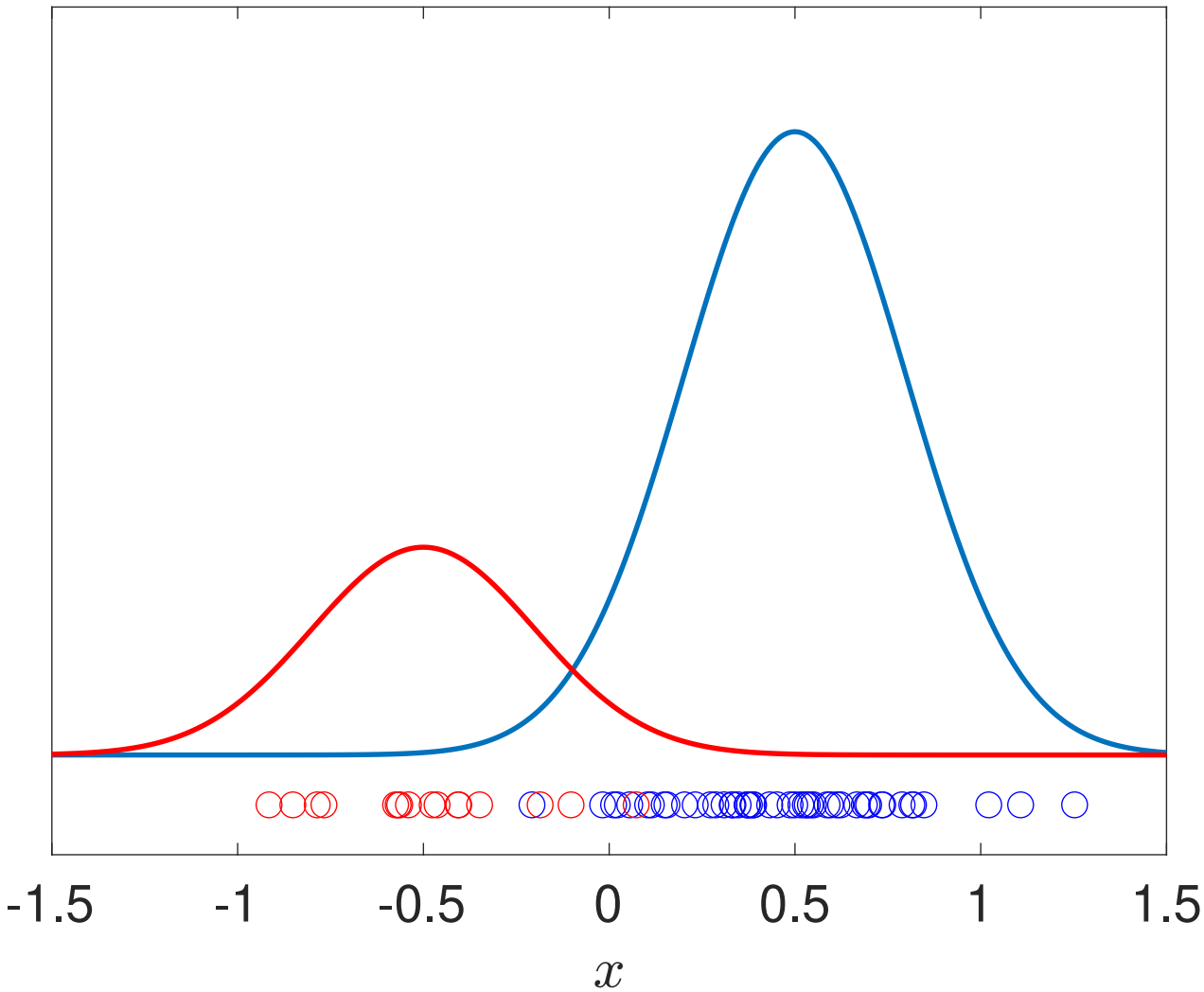
If there are K predictors, a Gaussian class density is expressed as:

$$p(\boldsymbol{x}|y = \text{o}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_\text{o}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_\text{o})^T \boldsymbol{\Sigma}_\text{o}(\boldsymbol{x}-\boldsymbol{\mu}_\text{o})}$$

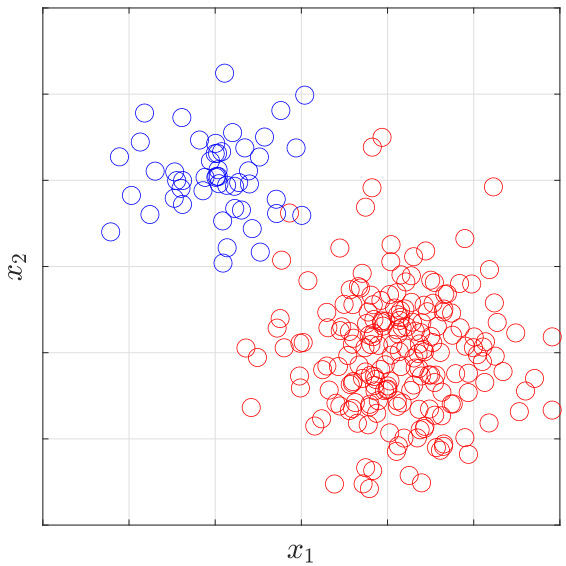
we $\boldsymbol{x} = [x_1, \ldots, x_K]^T$ contains all the predictors (note we have **not** prepended a 1), $\boldsymbol{\mu}_\text{o}$ is the **mean** and $\boldsymbol{\Sigma}_\text{o}$ is the **covariance matrix**.

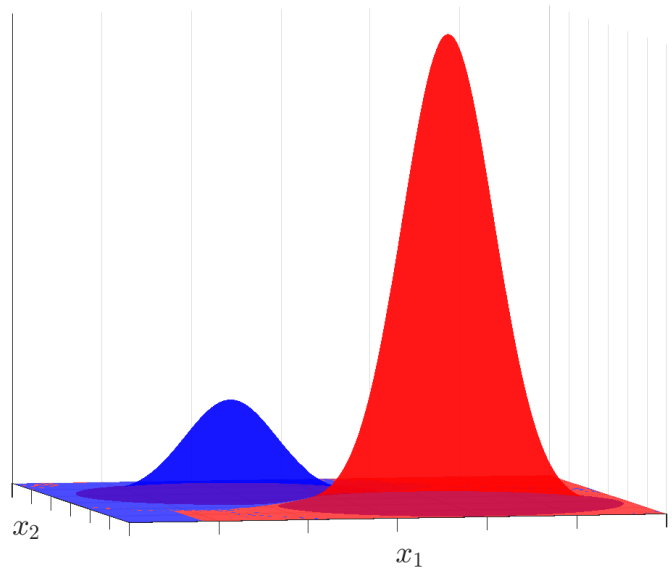
Similar expressions can be obtained for the o class densities.

Discriminant analysis: one predictor



Discriminant analysis: two predictors

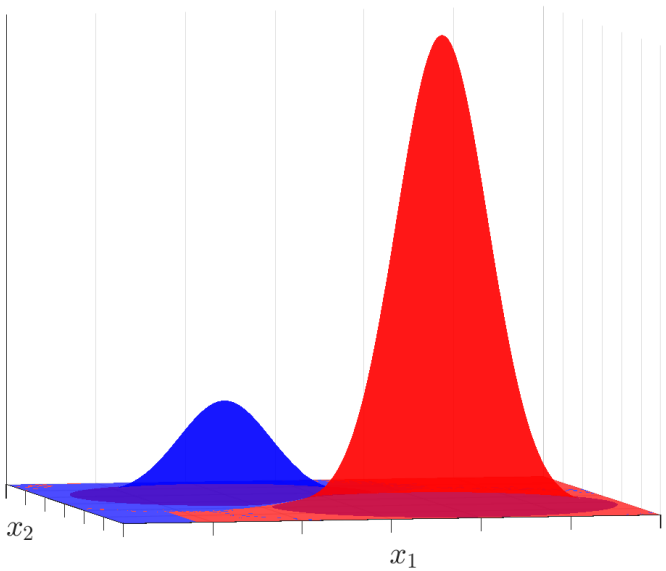


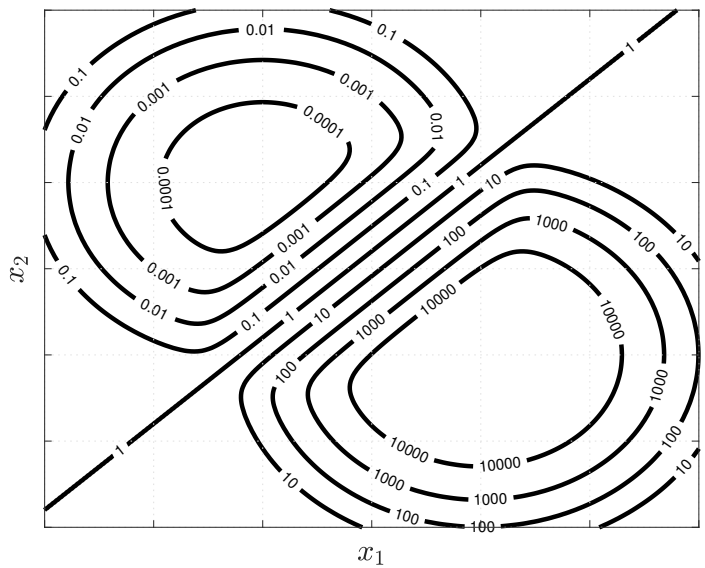


Linear and quadratic discriminant analysis

The boundary in discriminant analysis depends on the covariance matrices $\mathbf{\Sigma}_o$ and $\mathbf{\Sigma}_o$:

- If $\mathbf{\Sigma}_o = \mathbf{\Sigma}_o$ the boundary is linear. We call this scenario **linear discriminant analysis** (LDA).
- Otherwise, the boundary is quadratic. This is **quadratic discriminant analysis** (QDA).





Comparison of classifiers seen so far

- **Shape of boundaries**: Logistic regression and LDA build linear boundaries, QDA quadratic boundaries. kNN does not impose any particular shape.
- **Stability**: For a small number of samples, logistic regression can be very unstable, whereas LDA approaches produce stable solutions.
- **Outliers**: Logistic regression is robust against samples which lie very far from the boundary, LDA and QDA can be affected.
- **Multi-class**: Multi-class problems can be implemented easily in dicriminant analysis.
- **Prior knowledge**: Can be easily incorporated following Bayesian approaches.

Agenda

Is a high accuracy what we want?

Our notion of **quality** is defined by a **metric**, which allows us to rank different solutions. We have used two equivalent metrics for classifiers:

- **Accuracy**: Proportion of correctly classified samples.
- **Error rate**: Proportion of misclassified samples.

Note that both metrics are **blind to the class** that misclassified samples belong to.

However, is it the same misclassifying:

- A healthy patient and an ill patient, when deciding wether to administer some treatment?
- A good business and a bad business, when receiving a loan application?

If it is not, accuracy or error rate are no the quality metrics that we need.

A Bayesian extension

Consider a binary problem with two classes o and o, where:

- The cost of misclassifying a o sample is C_{o}.
- The cost of misclassifying a o sample is C_{o}.

The Bayes classifier achieves the highest accuracy by comparing:

$$\frac{P(y = \text{o} \,|\boldsymbol{x})}{P(y = \text{o} \,|\boldsymbol{x})} \lessgtr 1$$

and assigning the sample to o if > 1 or o if < 1.

The expected cost will be

- $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x})$, if our classifier labels the sample as o.
- $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x})$, if our classifier labels the sample as o.

A Bayesian extension

Consider the following posterior probabilities:

- $P(y = \text{o} \,|\boldsymbol{x}) = 0.9$,
- $P(y = \text{o} \,|\boldsymbol{x}) = 0.1$,

and misclassification costs:

- $C_{\text{o}} = 5000$ £,
- $C_{\text{o}} = 20$ £.

The Bayesian classifier would label sample $\boldsymbol{x}$ as o. Accordingly:

- 10 % of the time you would be misclassifying o samples.
- This would cost on average $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x}) = 5000 \times 0.1 = 500$ £.

What if we were to label sample $\boldsymbol{x}$ as o against the advice of the Bayes classifier? The accuracy would be lower (10 %), but the cost would be $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x}) = 20 \times 0.9 = 18$ £.

A Bayesian extension

To account for misclassification costs, we can use the following comparison instead:

$$\frac{C_{\text{o}} \times P(y = \text{o} \mid \boldsymbol{x})}{C_{\text{o}} \times P(y = \text{o} \mid \boldsymbol{x})} \lessgtr 1 \quad \text{or} \quad \frac{P(y = \text{o} \mid \boldsymbol{x})}{P(y = \text{o} \mid \boldsymbol{x})} \lessgtr \frac{C_{\text{o}}}{C_{\text{o}}}$$

A classifier that follows this strategy will **minimise the cost** defined by C_{o} and C_{o}, rather than the accuracy.

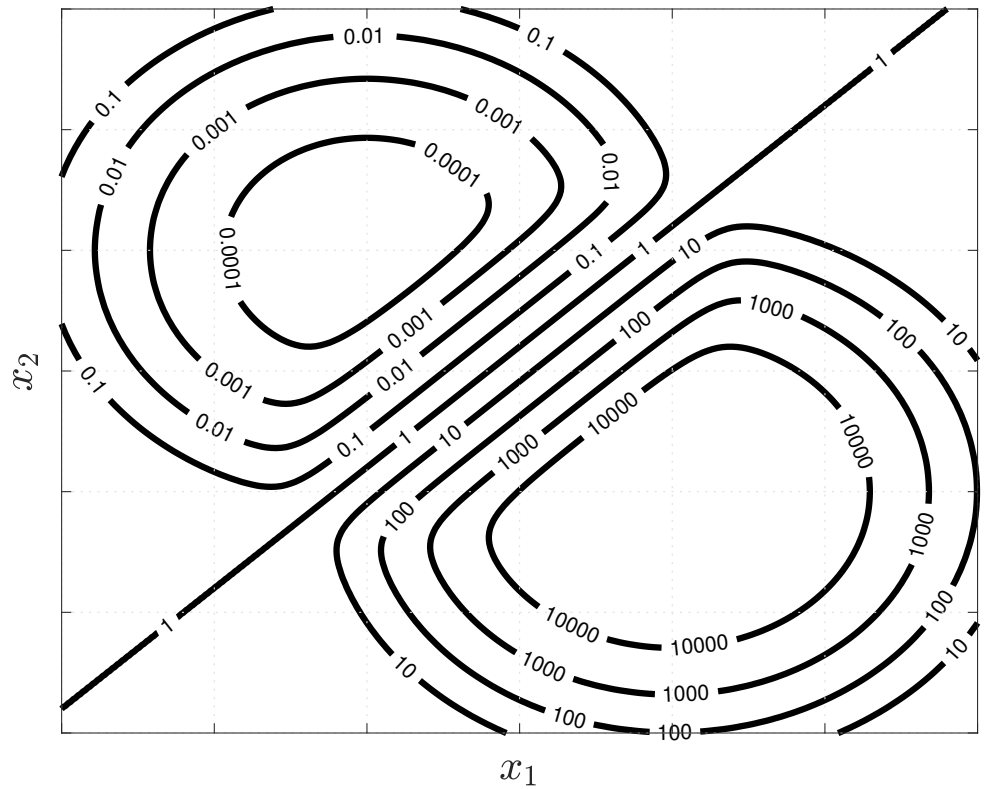
In general, our Bayesian extension will be expressed as

$$\frac{P(y = \text{o} \mid \boldsymbol{x})}{P(y = \text{o} \mid \boldsymbol{x})} \lessgtr T$$

where T is a threshold value.

A Bayesian extension

Changing the threshold value T changes the boundary of our classifier.



Confusion matrix

Accuracy and error rate are not be the most suitable quality metrics in **class-sensitive problems**, where the cost of misclassifying samples from different classes is different.

An alternative to using the **misclassification cost** for each class is to assess how well the classifier deals with each class separately. This is precisely the information that a **confusion** or **contingency matrix** shows.

Confusion matrix: Counting

A confusion matrix shows for each class, the number of samples:

- **Correctly classified**: diagonal
- **Misclassified**: non-diagonal

In the following confusion matrix, 3 o samples are misclassified as o, and 4 o samples are correctly classified. We can also learn that the dataset has 10 o samples, 20 o samples and 5 o samples and the **accuracy** is 24/35.

		Actual class		
		o	o	o
Predicted class	o	5	2	0
	o	3	15	1
	o	2	3	4

Confusion matrix: Rates

The confusion matrix can also show rates, defined as the proportion of samples from one class that are assigned to any other class.

This is useful when working with imbalanced datasets, where the counts might be misleading. The example confusion matrix below uses rates, instead of counts.

		Actual class		
		o	o	o
Predicted class	o	0.5	0.1	0
	o	0.3	0.75	0.2
	o	0.2	0.15	0.8

Confusion matrix: Detection problems

Many binary problems consider classes that represent the **presence** or **absence** of some property. For these problems, it is common to use the terms **positive** (presence) and **negative** (absence) for each class.

		Actual class	
		Positive	Negative
Predicted class	Positive	True positive	False positive
	Negative	False negative	True negative

In addition, we use the terms:

- True positive (TP) and true positive rate (TPR).
- False negative (FN) and false negative rate (FNR).
- False positive (FP) and false positive rate (FPR).
- True negative (TN) and true negative rate (TNR).

Confusion matrix: Detection example

Number of samples

	Actual	
Predicted	10	11
	2	9

Rates

	Actual	
Predicted	0.83	0.55
	0.17	0.45

- TP $= 10 \rightarrow$ TPR $= 10/12 = 0.83$
- FP $= 11 \rightarrow$ FPR $= 11/20 = 0.55$
- FN $= 2 \rightarrow$ FNR $= 2/12 = 0.17$
- TN $= 9 \rightarrow$ TNR $= 9/20 = 0.45$
- A $= (10+9)/32 = 19/32 = 0.59$
- E $= (11+2)/32 = 13/32 = 0.41$

Class-sensitive rates: Terminology

Error rate and accuracy are performance rates that do not allow us to investigate how a classifier treats each class. To do so, we can define other rates, such as the ones included in a confusion matrix.

In detection problems, the most commonly rates used are:

- **Sensitivity** (recall or true positive rate): TP/(TP+FN)
- **Specificity** (true negative rate): TN/(TN+FP)
- **Precision** (positive predictive value): TP/(TP+FP)

These rates can be used as **quality metrics**.

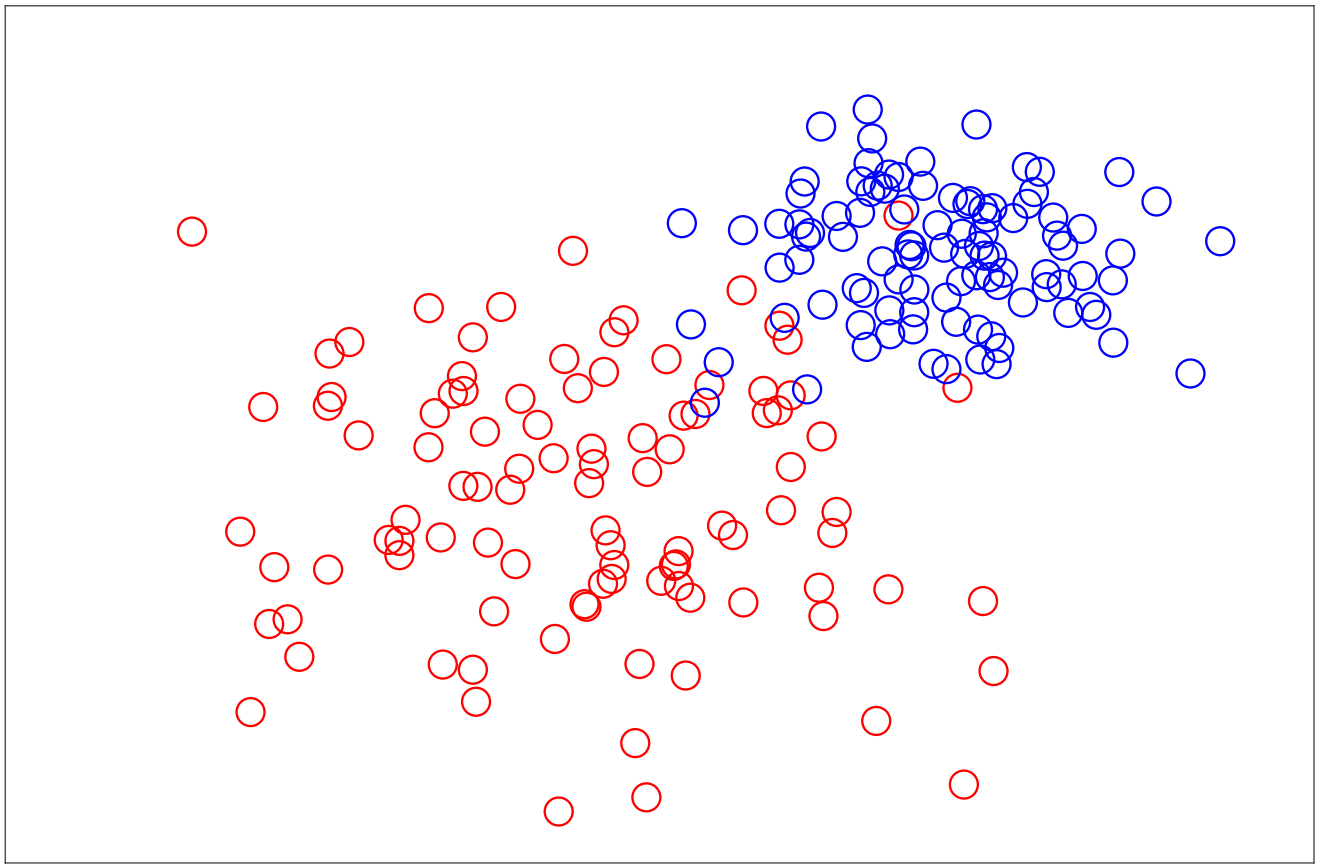
Class-sensitive rates rates: Example

	Actual	
Predicted	10	11
	2	9

- Sensitivity $= 10/12 = 0.83$
- Specificity $= 9/20 = 0.45$
- Precision $= 10/21 = 0.48$

Class-sensitive rates rates: Optimisation

If o is the positive class and o the negative class, obtain a linear boundary with the highest sensitivity and another with the highest specificity.



Confusion matrix: Optimisation

Improving one quality metric individually is easy. For instance, if we label every sample as positive, we would achieve a perfect sensitivity. The problem is that improving **one quality metric deteriorates others**.

Confusion matrix: Optimisation

Since improving the performance on class deteriorates the performance on the other, we usually consider simultanously **pairs of quality metrics**:

- Sensitivity and specificity.
- Precision and recall.

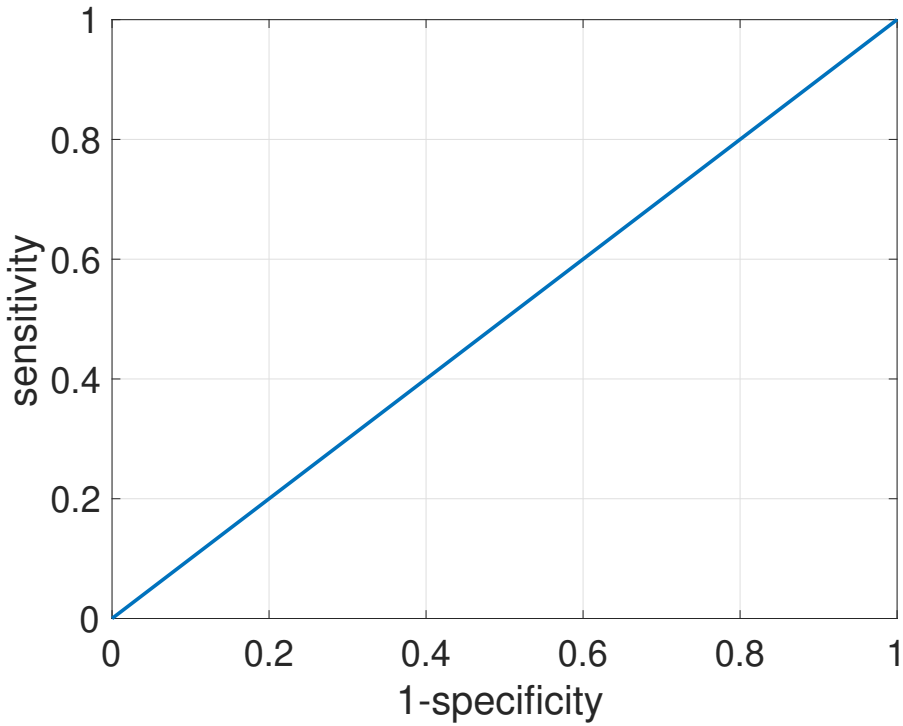
(*For instance, to travel to the UK you need to take a COVID test of 80 % sensitivity and 97 % specificity.*)

The F1-score is another widely used performance metric that provides an average between precision and recall:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

The ROC plane

The ROC (*receiver operating characteristic*) plane is used to represent the performance of a classifier in terms of its **sensitivity** and **1-specificity**.

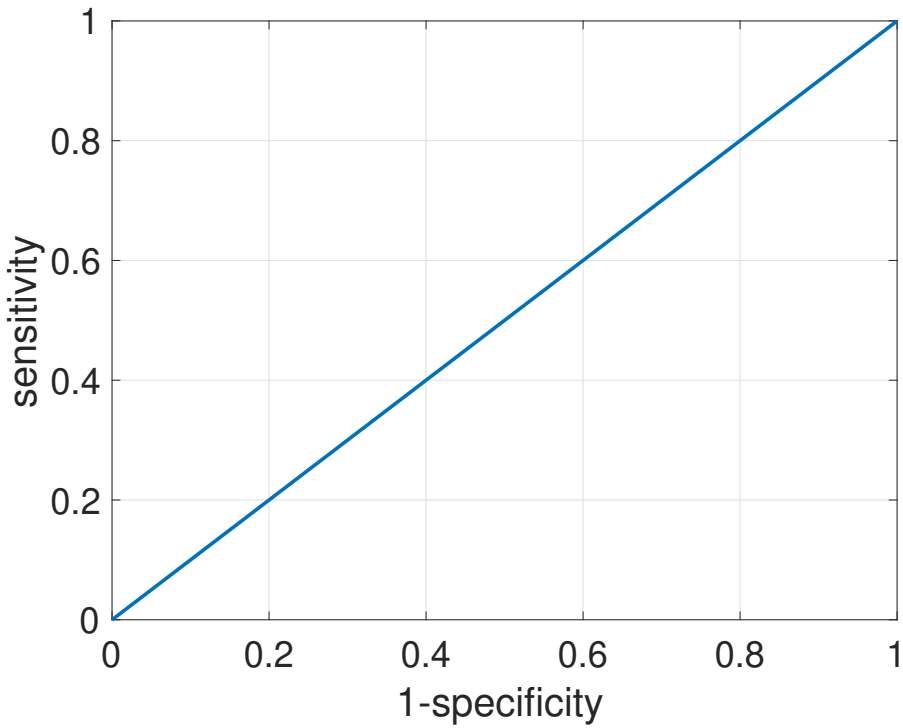


We would like the sensitivity to be close to 1 and the 1-specificity to be close to 0 (top left corner).

The ROC plane

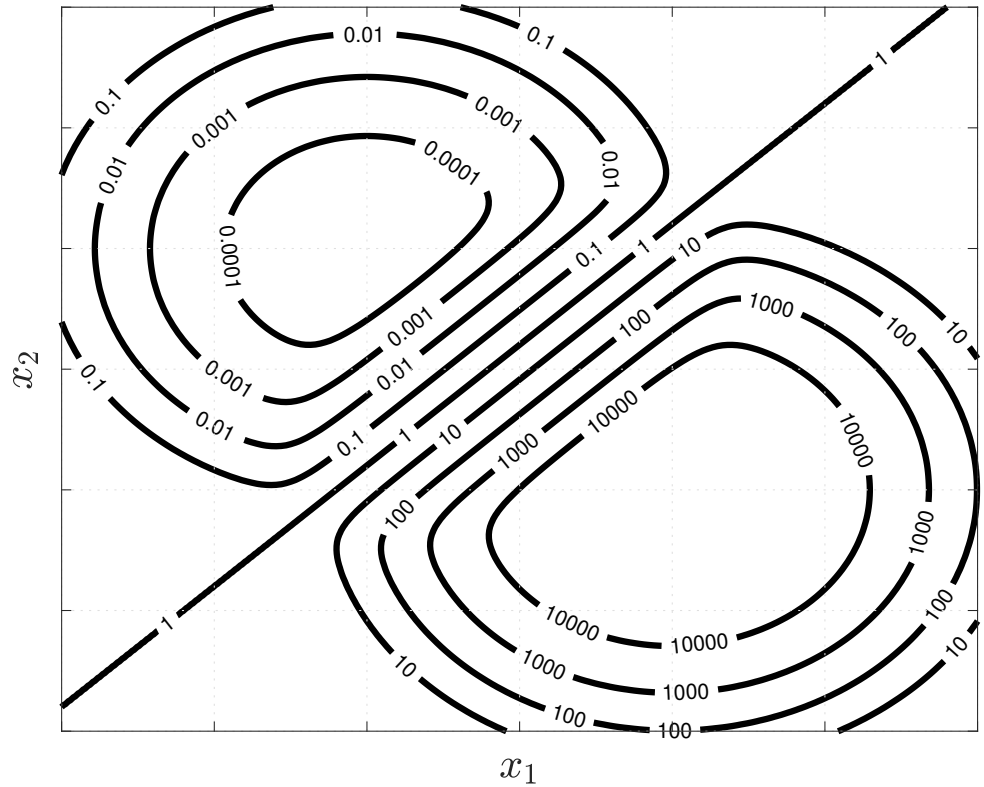
Note that we **cannot rank classifiers** using two metrics simultaneously.

The usual practice is to fix a minimum value for one of the metrics and optimise the other, for instance: obtain the highest sensitivity with a minimum specificity of 70 %.



Back to the decision boundary

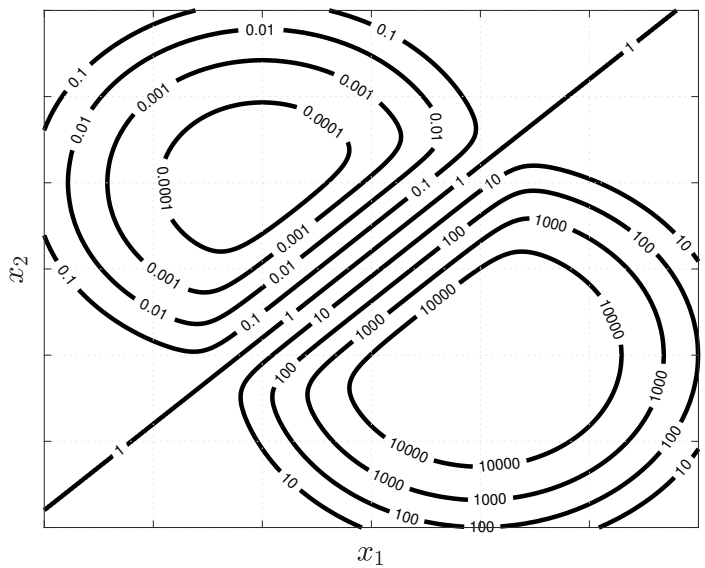
Our classifiers implement the comparison $\frac{P(y=\text{o}\,|\boldsymbol{x})}{P(y=\text{o}\,|\boldsymbol{x})} \lessgtr T$. For each value of T we have a different boundary that defines a different classifier.

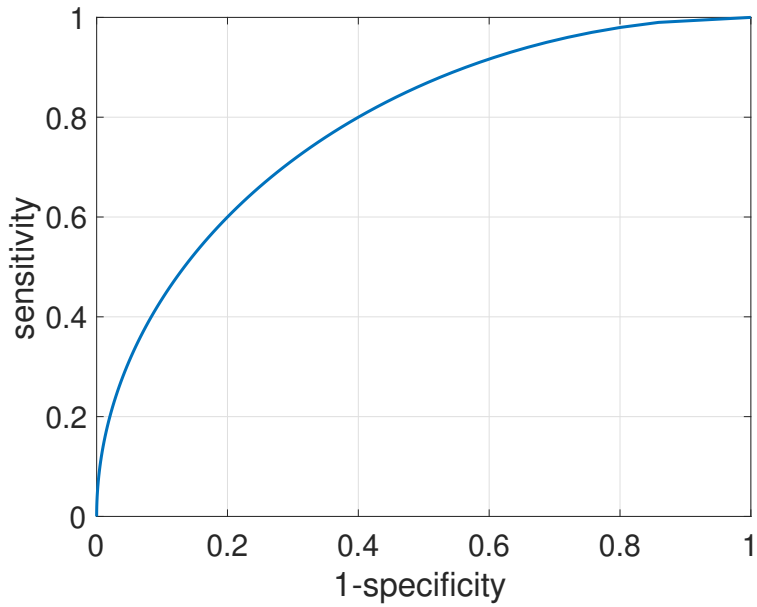


Hence, T can be **calibrated** to achieve your target performance.

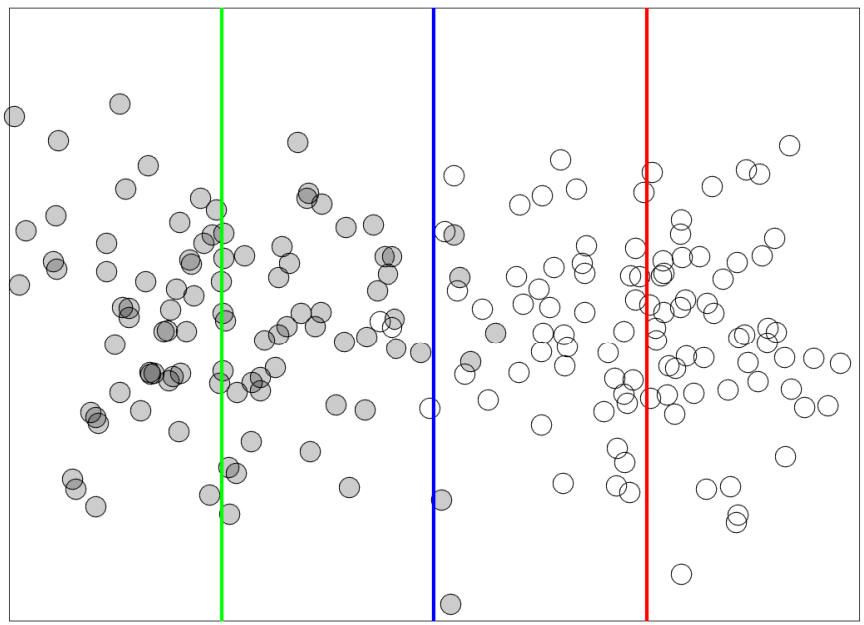
Calibration in the ROC plane: The ROC curve

We can represent in the ROC plane all the classifiers resulting from calibrating the threshold T. This is the **ROC curve**.





Calibration, boundaries and the confusion matrix

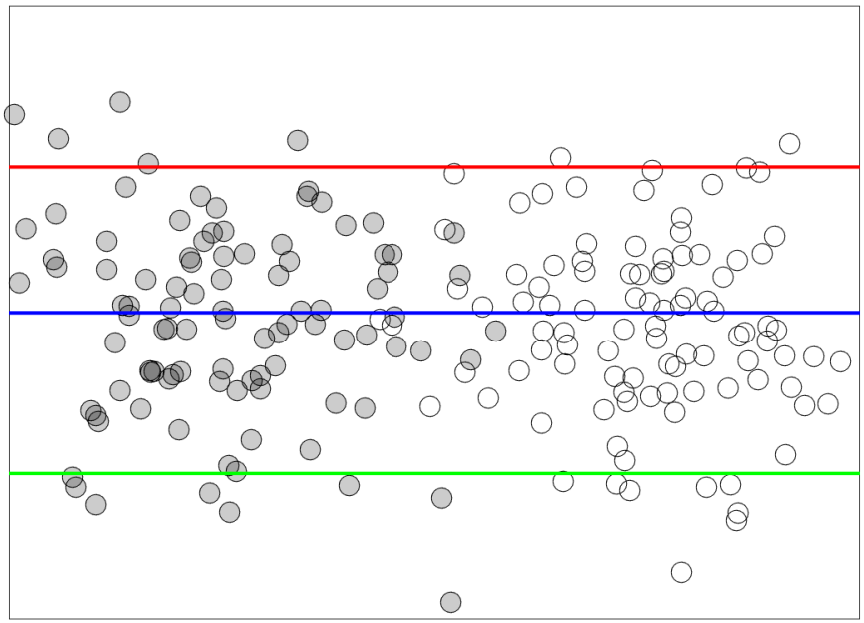


	Actual	
Predicted	1.00	0.50
	0	0.50

	Actual	
Predicted	0.95	0.05
	0.05	0.95

	Actual	
Predicted	0.50	0
	0.50	1.00

Calibration, boundaries and the confusion matrix



Predicted	Actual	
	0.05	0.05
	0.95	0.95

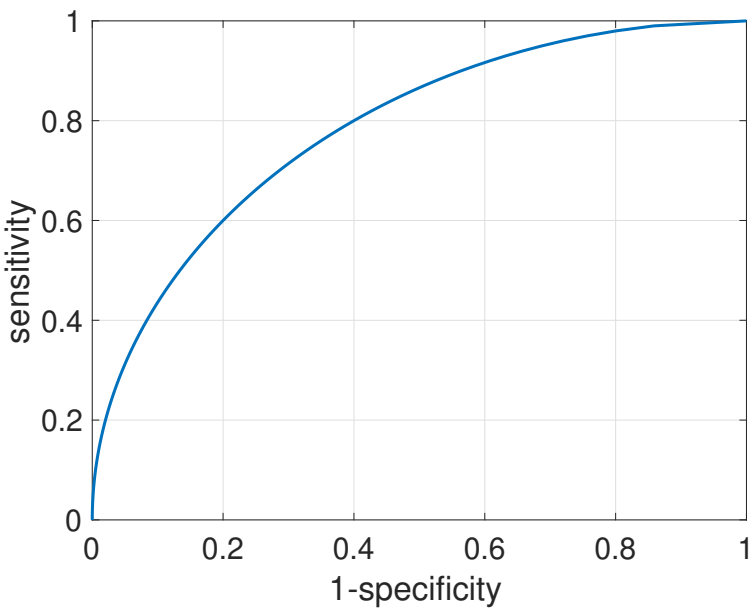
Predicted	Actual	
	0.50	0.50
	0.50	0.50

Predicted	Actual	
	0.92	0.92
	0.08	0.08

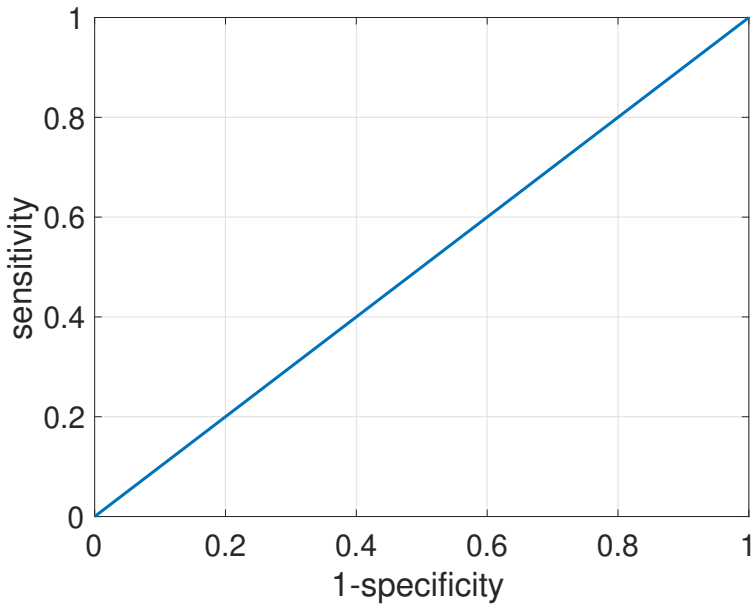
Calibration in the ROC plane: The AUC

The **area under the curve (AUC) is a measure of goodness** for a classifier that can be calibrated.

Good classifier (AUC $\approx$ 0.8)



Bad classifier (AUC = 0.5)



Good classifiers will have AUC close to 1, bad classifiers close to 0.5. Can you think of a classifier whose AUC<0.5?

Agenda

The Bayes classifier

Using data to build posterior probabilities

Classification performance: Beyond accuracy

Summary

The Bayes classifier

- The highest accuracy can be achieved comparing the posterior probabilities for each class and assigning a sample to the most probable class.
- The **Bayes classifier** is an ideal classifier that uses the **true posterior probabilities**.
- In general, we don't know the true posterior probabilities. In machine learning, classifiers can be seen as machines the use data to build posterior probabilities.

Beyond accuracy

- The accuracy does not tell us how a classifier treats each class, nor accounts for different costs in misclassifying samples from each class.
- If we know the misclassification costs, we can build classifiers that **minimise the global cost**, rather than maximising the accuracy.
- We can also define class-sensitive quality metrics, using the **confusion matrix**.
- Class-sensitive quality metrics are usually **conflicting**.
- The **ROC plane** allows to explore class-sensitive quality metrics.
- We always need a **test task** to evaluate the quality of a final classifier.

The best metric?

Don't take any quality metric for granted. Ask yourself: does this metric capture my notion of quality?

Think about the following class-sensitive classification scenarios. Which performance metrics would you use?

- Decision system in a bank offering loans.
- A security system to detect break-ins.
- A medical screening technique.
- Smoke alarm.