Queen Mary School Hainan
Queen Mary University of London

QHM5703 Principles of Machine Learning
Supervised learning: Classification II

Dr Nikesh Bajaj

Week 14: 8/9 Dec 2025

Agenda

Overview + Recap

So far we have covered:

- Regression: (label is a cont. variable)
- Methodology I: (three main tasks)
- Classification I: (label is a discreet variable)

You have also worked with

- Labs: Lab01 to Lab05
- Exercises: Regression and Methodology

Assessments so far

- Data Collection (4 %)
- Quiz (16 %)

Upcoming Assessments

- Mini-project (20 %)
- Final Exam (60 %)

Topics to be covered:

- Classification II
- Methodology II
- Structure Analysis (+lab)
- Density Estimation (+lab)
- Neural Networks and Deeplearning

Go to menti.com and use code:
DMT: 2429 5147
ICS: 1410 4001

Check QM+ Announcement about Feedback/SSLC

Queen Mary School Hainan
Queen Mary University of London

QHM5703 Principles of Machine Learning
Supervised learning: Classification II

Dr Nikesh Bajaj

Week 14: 8/9 Dec 2025

The best diagnostic machine

As the hospital's 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 (test 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

Go to menti.com and use code:

DMT: 2429 5147 — ICS: 1410 4001

Know your metrics!

The logistic model and kNN

We have introduced the **accuracy** and **error rate** as two convenient **target quality metrics**.

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

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

Note 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?

Cost vs quality

Regularisation provides an example where we use a notion of quality during training (E_{MSE+R}) that is different from the notion of quality during deployment (E_{MSE}).

Our goal is always to produce a model that achieves the highest **quality during deployment**. How we achieve it, is a different question.

In addition to using constraints that limit the risk of overfitting, the notion of training quality might be different from the deployment quality, because we cannot use the latter for training.

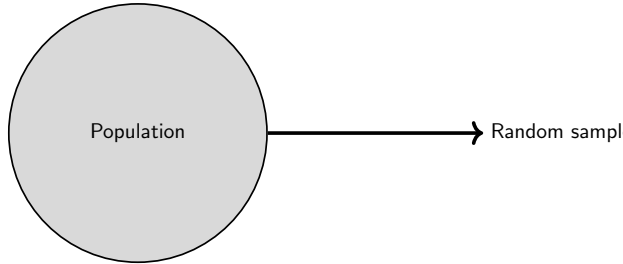
We usually call our notion of quality during training **cost** or **objective function**, to distinguish it from the **target quality metric**.

We will see examples where the notion of training cost is different from the notion of deployment quality. In many cases, we cannot use the notion of deployment quality during training.

Agenda

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.

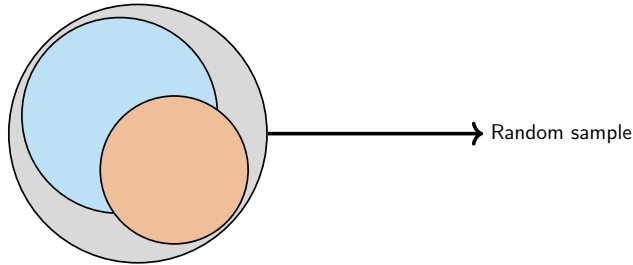


If we know nothing about our population, all we would flip a coin to classify and $A = 0.5$.

How can we use what we know about the population to improve our **classification accuracy**?

Prior probabilities: Predicting without predictors

Assume that we know that in 40 % of the population y = o,
whereas in the remaining 60 % y = o.



In this scenario, we know the **class priors**, namely $P(y = \text{o}) = 0.4$ and $P(y = \text{o}) = 0.6$.

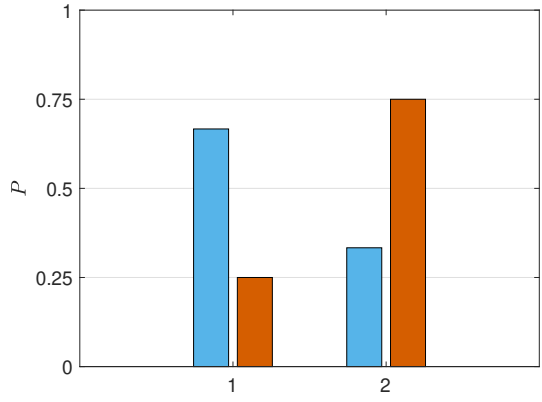
We would **always** choose o and $A = 0.6$. This is the **true accuracy**.

Likelihoods: The frequentist approach

What if we have access to the value of **another attribute** x and know **how frequently** a value of x appears within each class? These frequencies are known as **likelihoods**.

Assume x is binary and can take on the values 1 and 2:

$$
\begin{aligned}
P(x = 1|y = \text{o}) &= 1/4 \\
P(x = 2|y = \text{o}) &= 3/4 \\
P(x = 1|y = \text{o}) &= 2/3 \\
P(x = 2|y = \text{o}) &= 1/3
\end{aligned}
$$

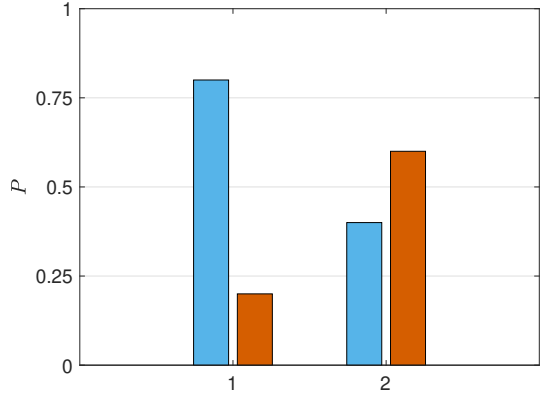


Given a sample where $x = 1$, we could compare $P(x = 1|y = \text{o})$ and $P(x = 1|y = \text{o})$ and decide that the sample belongs to o.

Posterior probabilities: The Bayesian approach

Posterior probabilities describe the probability that a sample belongs to a class given the value of its predictors. Do not confuse a posterior probability with a likelihood!

$$
\begin{aligned}
P(y = \text{o} | x = 1) &= 0.2 \\
P(y = \text{o} | x = 2) &= 0.6 \\
P(y = \text{o} | x = 1) &= 0.8 \\
P(y = \text{o} | x = 2) &= 0.4
\end{aligned}
$$



Given a sample where $x = 1$, we could compare $P(y = \text{o} | x = 1)$ and $P(y = \text{o} | x = 1)$ and decide that the sample belongs to o.

The Bayes classifier

The classifier that uses the **true posterior probabilities** is called the **Bayes classifier**.

The Bayes classifier uses the **odds ratio**:

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

If the ratio is greater than 1, the sample is o, if it is less, it is o.

The Bayes the classifier is the one that achieves the **highest accuracy**. In other words, you can **never beat** a Bayes classifier.

Agenda

Machine learning as statistical learning

To achieve the highest classification accuracy, we need to know the posterior probabilities. The question is, *where do we get them from*?

Machine learning classifiers use datasets to **estimate the posterior probabilities**. For instance:

- **Logistic models** use the logistic function (the classifier's *certainty*).
- **kNN** uses the fractions of neighbours that belong to each class.

The estimated posterior probabilities will in general be different from the **true posterior probabilities** and hence these classifiers will never beat the Bayes classifier.

Bayes rule and the odds ratio

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

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

The odds ratio can be expressed using the priors and the likelihoods:

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

Hence, the problem of **building posterior probabilities** is equivalent to the problem of **building priors and likelihoods**.

Building priors and likelihoods in machine learning: Example

x is discreet

y	x
o	1
o	1
o	2
o	2
o	2
o	1
o	1
o	1
o	1
o	1
o	1
o	2
o	2
o	2

Priors

Likelihoods

Posterior probabilities / Odd Ratio

Building priors and likelihoods in machine learning: Example

x is discreet

What does happen if Priors are changed?

y	x
o	1
o	1
o	2
o	2
o	2
o	1
o	1
o	1
o	1
o	1
o	1
o	2
o	2
o	2

Priors

Likelihoods

Posterior probabilities / Odd Ratio

Building priors and likelihoods in machine learning

In machine learning, we can use **data** to estimate the **priors** and the **likelihoods**. This is an **unsupervised problem** (more on this later).

Using a dataset to estimate the **priors** is very easy:

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

When the predictors are continuous, likelihoods are expressed as **class densities** $p(\boldsymbol{x}|y = \text{o})$ and $p(\boldsymbol{x}|y = \text{o})$. [*More in lecture: Density Estimation*]

In **discriminant analysis**, we assume that the class densities are **Gaussian** and use data to estimate:

- The means (μ_{o}, μ_{o}) and variances (σ^2_{o}, σ^2_{o}) for one predictor.
- The means ($\boldsymbol{\mu}_{\text{o}}$, $\boldsymbol{\mu}_{\text{o}}$) and covariance matrices ($\boldsymbol{\Sigma}_{\text{o}}$, $\boldsymbol{\Sigma}_{\text{o}}$) for multiple predictors.

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:

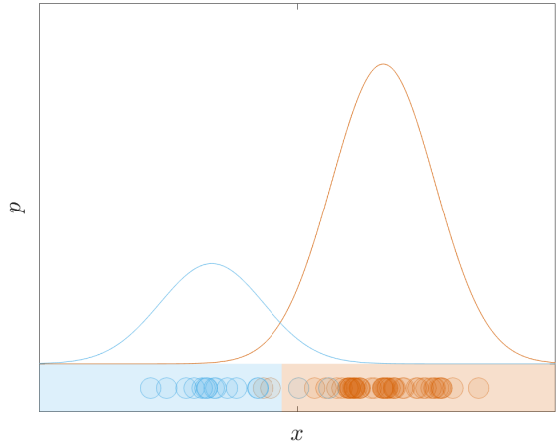
$$p(\boldsymbol{x}|y = \text{o}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_\text{o}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_\text{o})^T \boldsymbol{\Sigma}_\text{o}(\boldsymbol{x}-\boldsymbol{\mu}_\text{o})}$$

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**.

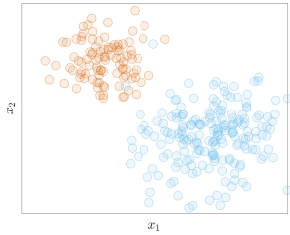
Similar expressions can be obtained for the o class densities.

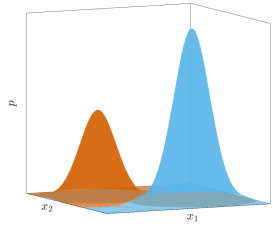
Discriminant analysis: one predictor

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

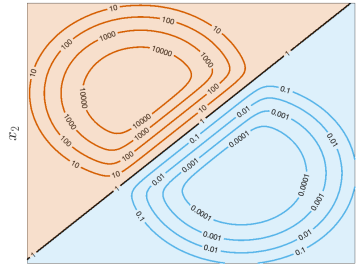


Discriminant analysis: two predictors





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



Linear and quadratic discriminant analysis

The shape of 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).

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 **unaware of 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.

Misclassification cost

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}.

Given the posterior probabilities $P(y = \text{o} \,|\boldsymbol{x})$ and $P(y = \text{o} \,|\boldsymbol{x})$, the expected cost will be:

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

Misclassification cost: Example

Consider the following posterior probabilities:

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

and misclassification costs:

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

If we label sample $\boldsymbol{x}$ as o (following Bayes):

- 90 % of the time we are right (it's o), 10 % wrong (it's o).
- The expected cost will be $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x}) = 5000 \times 0.1 = 500$ £.

If we label sample $\boldsymbol{x}$ as o (going against Bayes):

- 10 % of the time we are right (it's o), 90 % wrong (it's o).
- The expected cost will be $C_{\text{o}} \times P(y = \text{o} \,|\boldsymbol{x}) = 20 \times 0.9 = 18$ £.

Misclassification cost: A Bayesian extension

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

$$\frac{C_{\circ} \times P(y = \circ \,|\boldsymbol{x})}{C_{\circ} \times P(y = \circ \,|\boldsymbol{x})} \lessgtr 1 \quad \text{or} \quad \frac{P(y = \circ \,|\boldsymbol{x})}{P(y = \circ \,|\boldsymbol{x})} \lessgtr \frac{C_{\circ}}{C_{\circ}}$$

A classifier that follows this strategy will **minimise the expected cost**, rather than maximising the accuracy.

In general, our Bayesian extension will be expressed as

$$\frac{P(y = \circ \,|\boldsymbol{x})}{P(y = \circ \,|\boldsymbol{x})} \lessgtr T$$

where T is a threshold that we can control.

Note that we always use the odds ratio. If the misclassification costs change, we can use the same classifier **without retraining**.

A Bayesian extension: Calibration

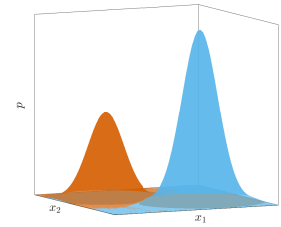
This map represents on the predictor space the odds ratio

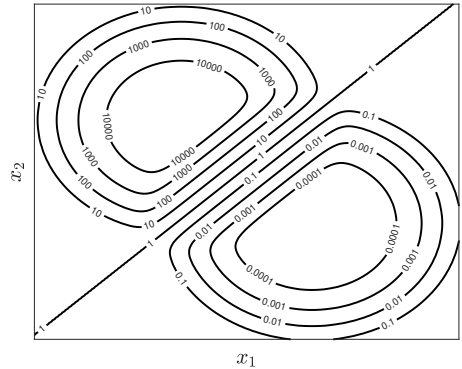
$$\frac{P(y = \text{o} \mid \boldsymbol{x})}{P(y = \text{o} \mid \boldsymbol{x})}$$

The decision regions are determined using the inequality

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

Changing the threshold value T (**calibration**) changes the decision boundary.





Per-class accuracies

Incorporating the misclassification cost in our formulation, we can build classifiers that **minimise the overall cost of classification**, rather than the overall misclassification error.

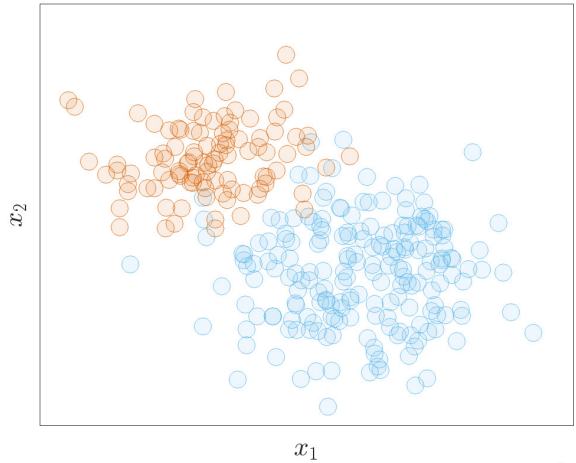
To understand **how each class is treated by the classifier**, we can obtain the **per-class accuracies**, which are defined on a dataset as:

$$\hat{A}_o = \frac{\#\text{true o predictions}}{\#\text{ o samples}} \quad \text{and} \quad \hat{A}_o = \frac{\#\text{true o predictions}}{\#\text{ o samples}}$$

The accuracy $\hat{A}$ does not distinguish between classes, whereas the per-class accuracies focus on each class separately.

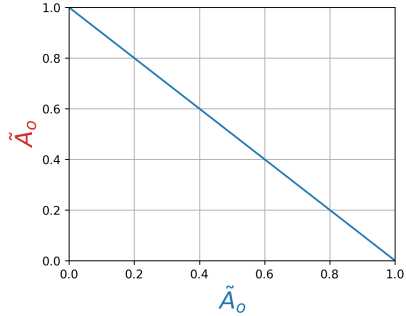
Per-class accuracies: Optimisation

Can we obtain a linear boundary that achieves at the same time the highest $\hat{A}_o$ and the highest $\hat{A}_o$?



The ROC plane: Simplified

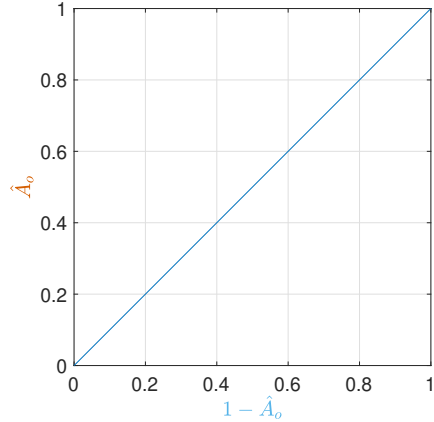
The ROC (*receiver operating characteristic*) plane is used to represent the performance of a classifier in terms of its per-class accuracies.



We would like $\hat{A}_o$ to be close to 1 and the $\hat{A}_o$ to be close to 1 (**top right** corner).

The ROC plane: Conventional

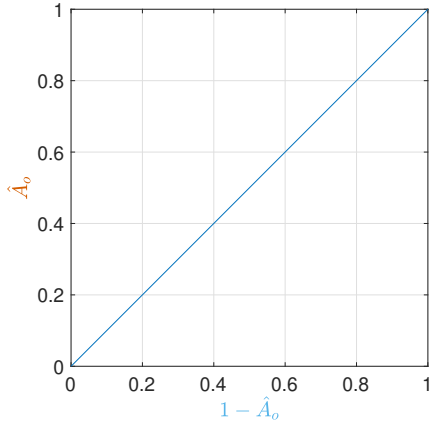
The ROC (*receiver operating characteristic*) plane is used to represent the performance of a classifier in terms of its per-class accuracies.



We would like $\hat{A}_o$ to be close to 1 and the $1 - \hat{A}_o$ to be close to 0 (top left corner).

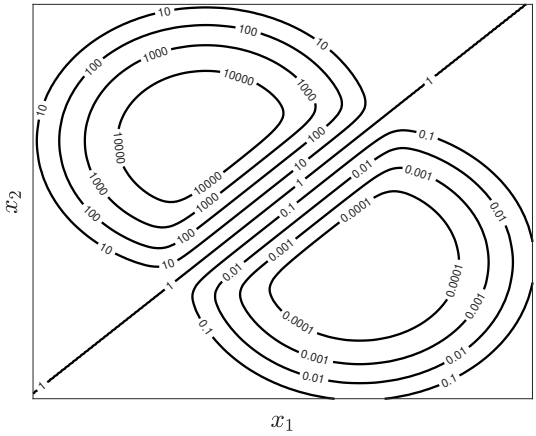
The ROC plane

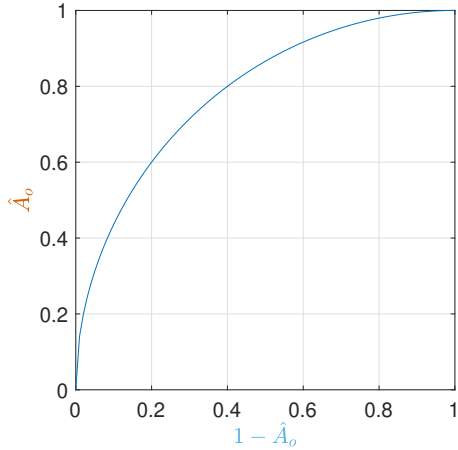
We **cannot rank classifiers** using two metrics simultaneously. We can fix a minimum value for one of the metrics and optimise the other, for instance: obtain the highest $\hat{A}_o$ with a minimum $\hat{A}_o$ of 70 %.



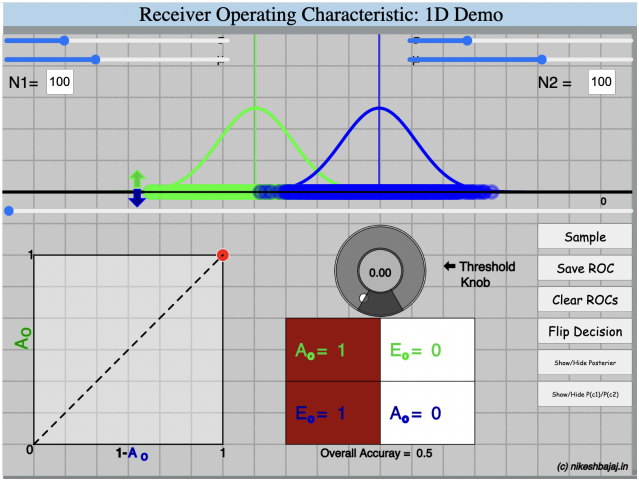
Calibrating in the ROC plane: The ROC curve

We can represent in the ROC plane **all the classifiers resulting from changing the threshold** T from 0 to ∞.





Calibrating in the ROC plane: DEMO + Quiz



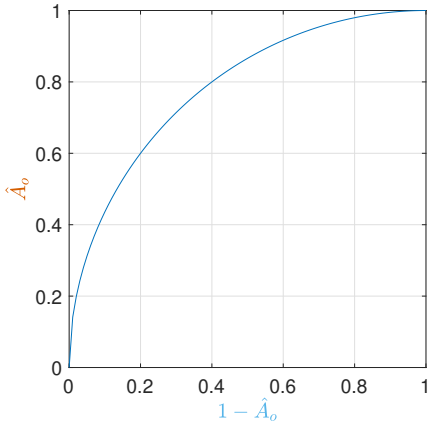
Quiz : based on Demo

Go to link: `https://nikeshbajaj.github.io/teaching/demos/ML/roc_v3.html`

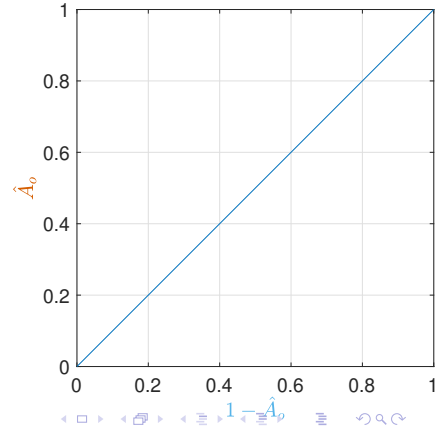
Calibrating in the ROC plane: The AUC

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

Good classifier (AUC ≈ 0.8)



Bad classifier (AUC = 0.5)



Confusion matrix: Counting

A confusion matrix provides a description of how samples within each class are classified:

- **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.

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

We can also learn that the dataset has 10 o samples, 20 o samples and 5 o samples and the **accuracy** is 24/35.

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. The diagonal values are in fact the **per-class accuracies**.

		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

Rates are useful when working with imbalanced datasets, where the counts might be misleading.

Agenda

What is a detection problem?

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.

Examples include:

- Radar system.
- Diagnostic tool.
- Document retrieval.

Detection problems appear in many scientific and engineering fields. They use the same underlying concepts however, they sometimes **name these concepts differently**.

Detection problems: Confusion matrix

As binary problems, the confusion matrix is 2×2. Their cells are named following the convention to name each class as positive or negative.

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

Depending on whether we represent counts or rates, we have:

- 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 $= (11+2)/32 = 13/32 = 0.41$

Detection problems: Terminology

Other names for the rates in a confusion matrix which are specific to detection problems include:

- **True positive rate**: Sensitivity or recall.
- **True negative rate**: Specificity

In addition, the **precision** or positive predictive value is defined as TP/(TP+FP).

		Actual	
Predicted		10	11
		2	9

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

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

Detection problems: Optimisation

Since improving the performance on one class can deteriorate the performance on the other, we sometimes consider **pairs of quality metrics** simultaneously:

- Sensitivity and specificity.
- Precision and recall.

(*For instance, during the pandemic, to travel to the UK we needed 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:

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

Question [HW: Take your time and do it properly]

- **Case 1**: You have a total 550 samples, from which 220 are class A, rest are class B. Your trained model is predicting 250 samples as class A, out of which 180 samples are actually from class A.
- **Case 2**: You have total 380 samples, from which 40 are class A, rest are class B. Your trained model is predicting all the samples as class B.

Before computing any metric, think about the model and its performance. What do you think, your trained model is doing in each case?

Now Compute for each case:

- Accuracy
- TP, TN, FP, FN
- Sensitivity, Specificity, Precision, Recall, and F1-score

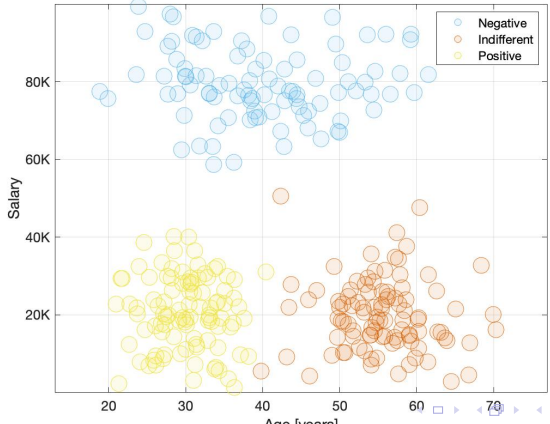
Wait a minute! Which class did you consider as Positive? Actually it was class B that was positive.
Which metric do you think, is more appropriate to reflect the goodness of your model?:

Agenda

Muticlass classifier

How do we build multi-class classifier?

- One-vs-One
- All-vs-all
- **One-vs-all**



Agenda

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 that use **data to build posterior probabilities**.

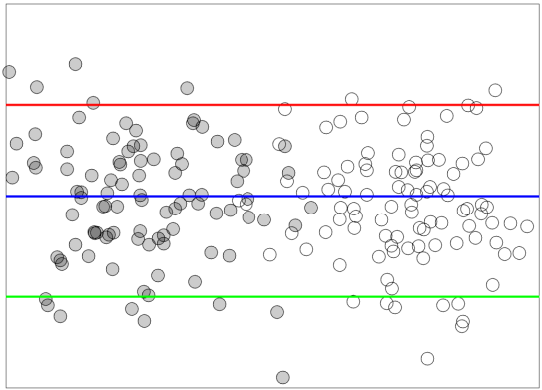
Comparison of classifiers seen so far

- **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 DA approaches produce stable solutions.
- **Outliers**: Logistic regression is robust against samples which lie very far from the boundary, LDA and QDA can be affected.
- **Multiclass**: Multiclass problems can be implemented easily in dicriminant analysis.
- **Prior knowledge**: Can be easily incorporated following Bayesian approaches.

Beyond accuracy

- The accuracy does not tell us how a classifier treats each class, nor accounts for different misclassification costs.
- If we know the misclassification costs, we can build classifiers that **minimise the global cost**, rather than maximising the accuracy.
- We can also define **per-class accuracies** and the **confusion matrix** to investigate how each class is treated.
- Per-class quality metrics are usually **conflicting**.
- The **ROC plane** allows to compare classifiers using per-class quality metrics.

Calibration and the confusion matrix: Example

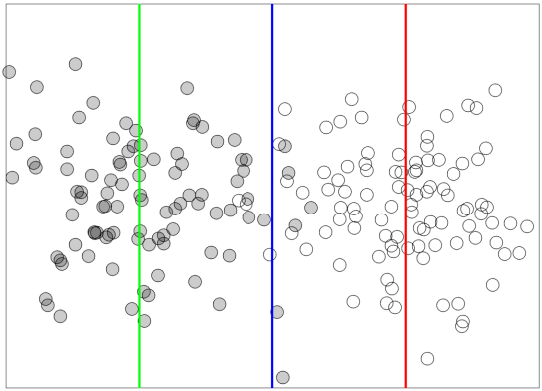


	Actual	
Predicted	0.05	0.05
	0.95	0.95

	Actual	
Predicted	0.50	0.50
	0.50	0.50

	Actual	
Predicted	0.92	0.92
	0.08	0.08

Calibration and the confusion matrix: Example



	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

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.

Queen Mary
University of London