

# Bayesian Models

Payam Barnaghi  
 Department of Brain Sciences &  
 School of Convergence Science in Human and Artificial Intelligence  
 Imperial College London  
 January 2025

I

1

## Probability of a union of two events

— Given two events, A and B, we define the probability of A or B as follows:

$$\begin{aligned} p(A \vee B) &= p(A) + p(B) - p(A \wedge B) \\ &= p(A) + p(B) \text{ if } A \text{ and } B \text{ are mutually exclusive} \end{aligned}$$



2

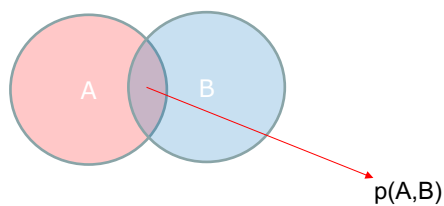
2

## Conditional probability

IMPERIAL

- We define the conditional probability of event A, given that event B is true, as follows:

$$p(A|B) = \frac{p(A, B)}{p(B)} \text{ if } p(B) > 0$$

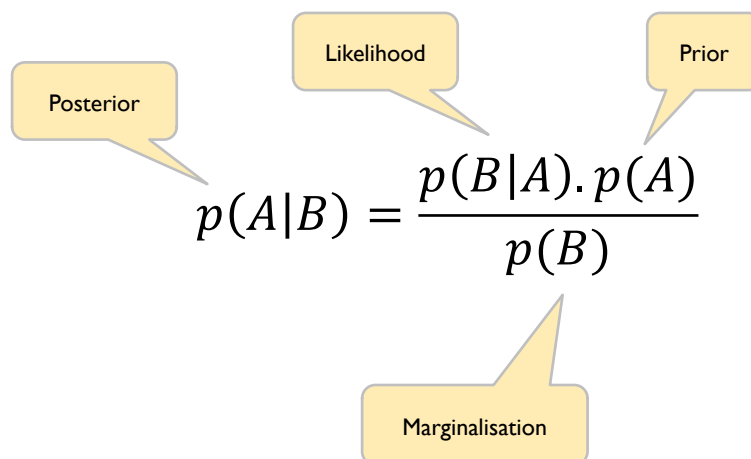


3

3

## The Bayes rule

IMPERIAL



4

4

## Bayes theorem

IMPERIAL

- Combining the definition of conditional probability with the product and sum rules yields the Bayes rule, also called the Bayes theorem:

$$p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)} = \frac{p(X = x)p(Y = y|X = x)}{\sum_{x'} p(X = x')p(Y = y|X = x')}$$

5

5

## Bayesian Classification

IMPERIAL

- Naive Bayes classifiers are built on Bayesian classification methods.
- These rely on Bayes's theorem, which describes the relationship of conditional probabilities of statistical quantities.
- In Bayesian classification, we are interested in finding the probability of a label given some observed features, which we can write as  $P(\text{label} | \text{features})$ .

Source: Python Data Science Handbook by Jake VanderPlas.

6

6

## Naïve Bayes

IMPERIAL

- We call it Naïve because it assumes that all the features are independent. In practice, this is not always true – but this assumption allows us to apply the Bayesian theorem:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

7

7

## Assumptions in Naïve Bayes

IMPERIAL

- We assume that the features are:
  - Independent given the outcome
- The assumption is not correct in the real world, but the models we can build can often be useful in real-world practice (obviously with caution).

8

8

## Bayes rule

IMPERIAL

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

- $y$  is our outcome/target (e.g. if someone has a type of disease)
- $X$  represents the samples; each features is shown as  $x_i$
- $X = \{x_1, x_2, x_3, \dots, x_n\}$  are the features.

9

9

## Determining Naïve Bayes

IMPERIAL

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

Because we have assumed features are conditionally independent, we can say:

$$\begin{aligned} P(y|x_1, x_2, x_3, \dots, x_n) &= \frac{P(x_1|y) \cdot P(x_2|y) \cdot P(x_3|y) \dots P(x_n|y) \cdot P(y)}{P(X)} \\ &= \frac{P(x_1|y) \cdot P(x_2|y) \cdot P(x_3|y) \dots P(x_n|y) \cdot P(y)}{P(x_1)P(x_2)P(x_3) \dots P(x_n)} \end{aligned}$$

10

10

Let's simplify

IMPERIAL

$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^D p(x_j|y)$$

11

11

Naïve Bayes: example

IMPERIAL

P (disease = 1) given [0,1]:

$$P(\text{Disease} = 1 | \text{Feature1} = 0, \text{Feature2} = 1) = \frac{P(\text{Feature1} = 0 | \text{Disease} = 1) \cdot P(\text{Feature2} = 1 | \text{Disease} = 1) \cdot P(\text{Disease} = 1)}{P(\text{Feature1} = 0, \text{Feature2} = 1)}$$

	High BP	TBI	Disease
Patient #01	1	0	0
Patient #02	1	1	1
Patient #03	1	0	0
Patient #04	0	0	0
Patient #05	1	0	1
Patient #06	0	0	1
Patient #07	0	0	0
Patient #08	0	1	1
Patient #09	0	1	0
Patient #10	1	0	0

TBI: History of Traumatic Brain Injury; BP: Blood Pressure

12

12

## Algorithm for Naïve Bayes (with binary features)

IMPERIAL

---

```

1  $N_c = 0, N_{jc} = 0;$ 
2 for  $i = 1 : N$  do
3    $c = y_i$  // Class label of  $i$ 'th example;
4    $N_c := N_c + 1$  ;
5   for  $j = 1 : D$  do
6     if  $x_{ij} = 1$  then
7        $N_{jc} := N_{jc} + 1$ 
8  $\hat{\pi}_c = \frac{N_c}{N}, \hat{\theta}_{jc} = \frac{N_{jc}}{N}$ 

```

13

13

## What about continuous variables?

IMPERIAL

- In the case of real-valued features, we can use the Gaussian distribution:

$$p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^D \mathcal{N}(x_j | \mu_{jc}, \sigma_{jc}^2)$$

where  $\mu_{jc}$  is the mean of feature  $j$  in objects of class  $c$ , and  $\sigma_{jc}^2$  is its variance.

14

14

## Bayesian decision methods

IMPERIAL

- Suppose we have an input vector  $\mathbf{x}$  with a corresponding vector  $\mathbf{y}$  of target variables.
- Our goal is to predict  $\mathbf{y}$  given a new value for  $\mathbf{x}$ .
- For a regression problem,  $\mathbf{y}$  will contain continuous (real-valued) variables, and for a classification problem,  $\mathbf{y}$  will represent class labels.

15

15

## Joint probability

IMPERIAL

- The joint probability distribution  $p(\mathbf{x}, \mathbf{y})$  provides a summary of the uncertainty associated with these variables.
- Determining  $p(\mathbf{x}, \mathbf{y})$  from a set of training data is also referred to as *inference*.
- Consider a medical diagnosis example: We have taken an MRI scan of a patient to determine whether the patient has a particular type of neurological disorder.

16

16



## Medical diagnosis example

IMPERIAL

- In the previous example, the input vector  $x$  is the set of pixels (or engineered features) from the scan.
- $y$  will represent the presence of the disease; for the presence of the disease, we show it, for example, with  $C_1$ , and for the absence of the disease, we note it as  $C_2$ .
- The general inference problem then involves determining the joint distribution  $p(x, C_k)$

17

17

## Example- applying Bayesian theorem

IMPERIAL

$$p(C_k|x) = \frac{p(x|C_k) p(C_k)}{p(x)}$$

$p(C_k)$  is our prior

$p(C_k|x)$  is our posterior

Posterior = prior  $\times$  Likelihood / evidence

Posterior  $\propto$  prior  $\times$  Likelihood

18

18

## Bayesian inference

IMPERIAL

$$p(\theta|data) = \frac{p(data|\theta) \times p(\theta)}{p(data)}$$

$\theta$  represents the unknown parameter(s) that we want to estimate

19

19

## Likelihood

IMPERIAL

- $p(data|\theta)$  is called likelihood.
- This means the probability of generating a particular sample if the parameter of our model was equal to  $\theta$ .
- For example, if our data includes samples from a population with features that indicate if they had high blood pressure and if they had a history of TBI (X) associated with the target variable as having Alzheimer's disease or not (y); then
- Then the likelihood would present, if someone had Alzheimer's disease, what is the probability that they had high blood pressure and/or a history of TBI (depending on what likelihood we are interested in).

20

20

## Maximum likelihood\*

IMPERIAL

- In principle, we define the likelihood function and then try to calculate the parameter values that maximise the likelihood of obtaining our data.
- In other words, in the previous example,  $p(\text{data} \mid \theta)$ , we assumed that we know the parameter  $\theta$ , but what if we try to use a method to identify what value(s) for  $\theta$  would maximise the likelihood of obtaining the data?
- For example, in the case of a disease, we may not know the prevalence of the disease in advance in the population.

21

21

## Estimating the maximum likelihood\*

IMPERIAL

- For example, let's assume in a sample of 100 individuals with a neurological disorder, we have observed that 10 of them have a specific symptom.
- This means that the overall likelihood of having this symptom is given by:

$$\mathcal{L}(\theta \mid X = 10, N = 100) = \binom{100}{10} \theta^{10} (1 - \theta)^{90}$$

- Note that since we are varying  $\theta$  and assuming data is constant, this is a likelihood and not a probability.

22

22

## Estimating the maximum likelihood\*

IMPERIAL

- We are now interested in finding the value of  $\theta$  that maximises the likelihood function.
- This is called maximum likelihood.
- This is often used in probabilistic machine learning models to find the parameters for the model that will maximise the likelihood function and find the optimum parameters that we can use in our model to make predictions in the future.
- Reminder:

$$\text{Posterior} \propto \text{Likelihood} \times \text{prior}$$

$$p(\theta|data) = \frac{p(data|\theta) \times p(\theta)}{p(data)}$$

23

23

## Repeating our example\*

IMPERIAL

- For example, let's assume in a sample of 100 individuals with a neurological disorder, we have observed that 10 of them have a specific symptom.
- This means that the overall likelihood of having this symptom is given by:

$$\mathcal{L}(\theta|X = 10, N = 100) = \binom{100}{10} \theta^{10} (1 - \theta)^{90}$$

- How to find what value for theta ( $\theta$ ) will maximise this?

24

24

## Log likelihood\*

IMPERIAL

- Before we continue with the next few slides:
- I don't expect you to know or use the maths discussed here- this is just added for your information and for those who might be interested in pursuing Bayesian methods further.
- Most modern libraries will train/learn your model and optimise the parameters for you automatically, including the models we will use in our lab experiments.

25

25

## Log likelihood\*

IMPERIAL

$$\mathcal{L}(\theta|X = 10, N = 100) = \binom{100}{10} \theta^{10} (1 - \theta)^{90}$$

- How to find what value for theta will maximise this?
- These three rules are going to help us:

$$\log(ab) = \log(a) + \log(b)$$

$$\log(a^b) = b \log(a)$$

$$\frac{d}{dx} (\log_a x) = \frac{1}{x \ln a}$$

26

26

## Maximising the Log likelihood\*

**IMPERIAL**

- To maximise the likelihood, calculate log-likelihood, take the derivative and then set it equal to zero.
- We calculate the log-likelihood for simplicity of working with sums (+) instead of multiplications (x) [needed for the step below]
- We take the first derivative and set it to zero because the first derivative will give us the slope of change;
- And setting it to zero will give you the point that the point that the function is maximised.

27

27

## Why?

**IMPERIAL**

- Why do we take the first derivative and set it to zero?

28

28

## Maximising the Log likelihood\*

IMPERIAL

$$\begin{aligned}\mathcal{L}(\theta|X = 10, N = 100) &= \binom{100}{10} \theta^{10} (1 - \theta)^{90} \\ \log(ab) &= \log(a) + \log(b) \\ \log(a^b) &= b \log(a) \\ \frac{d}{dx}(\log_a x) &= \frac{1}{x \ln a} \\ \log \mathcal{L}(\theta|X = 10, N = 100) &= \log \binom{100}{10} + \log \theta^{10} + \log(1 - \theta)^{90} \\ \frac{\partial l}{\partial \theta} &= \ln a \left( \frac{10}{\theta} - \frac{90}{1 - \theta} \right) = 0 \\ \hat{\theta} &= 1/10\end{aligned}$$

\*\* We did not define any specific base for the logarithm as it is not important here; overall,  $\text{Log}_b x = \ln(x)/\ln(b)$

29

29

## Naïve Bayes in python

IMPERIAL

- It required a small amount of training data to estimate the necessary parameters.
- Naive Bayes learners and classifiers can be very fast compared to more sophisticated methods.
- On the flip side, although naïve Bayes is known as a decent classifier, it is **known to be a bad estimator, so the probability** outputs from `predict_proba` (in Python scikit learn) are not to be taken too seriously.

Source: [https://scikitlearn.org/stable/modules/naive\\_bayes.html](https://scikitlearn.org/stable/modules/naive_bayes.html)

30

30

## Naïve Bayes in python

IMPERIAL

- Gaussian Naive Bayes:
- **GaussianNB** implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

- The parameters  $\sigma_y$  and  $\mu_y$  are estimated using maximum likelihood.

Source: [https://scikitlearn.org/stable/modules/naive\\_bayes.html](https://scikitlearn.org/stable/modules/naive_bayes.html)

31

31

## Gaussian Naive Bayes

IMPERIAL

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.naive_bayes import GaussianNB
>>> X, y = load_iris(return_X_y=True)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
>>> gnb = GaussianNB()
>>> y_pred = gnb.fit(X_train, y_train).predict(X_test)
>>> print("Number of mislabeled points out of a total %d points : %d"
...      % (X_test.shape[0], (y_test != y_pred).sum()))
Number of mislabeled points out of a total 75 points : 4
```

Source: [https://scikitlearn.org/stable/modules/naive\\_bayes.html](https://scikitlearn.org/stable/modules/naive_bayes.html)

32

32



## Multinomial Naïve Bayes

IMPERIAL

- **MultinomialNB** implements the naïve Bayes algorithm for multinomially distributed data and is one of the two classic naïve Bayes variants used in text classification.
- The distribution is parametrised by vectors  $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$  for each class  $y$ , where  $n$  is the number of features and  $\theta_{yi}$  is the probability  $P(x_i|y)$  of feature  $i$  appearing in a sample belonging to class  $y$ .
- The parameters  $\theta_y$  is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

Source: [https://scikitlearn.org/stable/modules/naive\\_bayes.html](https://scikitlearn.org/stable/modules/naive_bayes.html)

33

33

## Multinomial Naïve Bayes\*

IMPERIAL

- The parameters  $\theta_y$  is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where  $N_{yi} = \sum_{x \in T} x_i$  is the number of times feature  $i$  appears in a sample of class  $y$  in the training set  $T$ , and  $N_y = \sum_{i=1}^n N_{yi}$  is the total count of all features for class  $y$ .

The smoothing priors  $\alpha \geq 0$  accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting  $\alpha = 1$  is called Laplace smoothing, while  $\alpha < 1$  is called Lidstone smoothing.

Source: [https://scikitlearn.org/stable/modules/naive\\_bayes.html](https://scikitlearn.org/stable/modules/naive_bayes.html)

34

34

## Evaluation metrics

35

## Precision and Recall

- Precision defines what proportion of positive identifications have actually been correct.

$$Precision = \frac{TP}{TP + FP}$$

TP = True Positive

FP = False Positive

Example: Calculate the Precision

True Positives: 25	False Positives: 10
False Negatives: 15	True Negatives: 80

36

36

## Recall

IMPERIAL

- Recall defines the proportion of actual positives that were identified correctly.

$$\text{Recall} = \frac{TP}{TP + FN}$$

TP = True Positive

FN = False Negative

Example: Calculate the Recall

True Positives: 25	False Positives: 10
False Negatives: 15	True Negatives: 80

37

37

## Precision and Recall

IMPERIAL

How decision threshold could affect Precision and Recall.

This can be very important in medical applications.

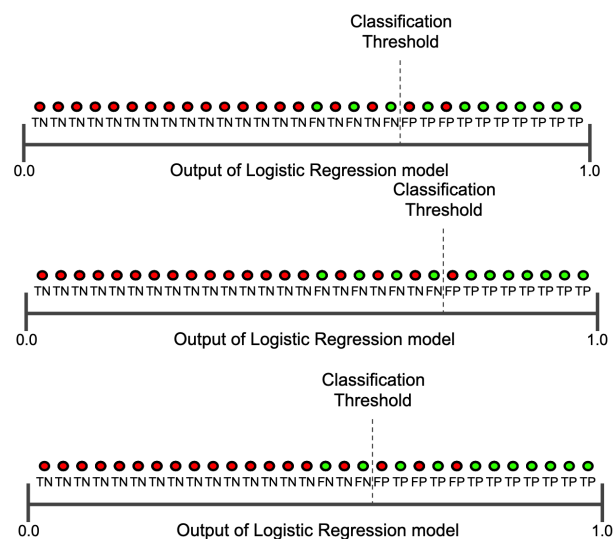


Image source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

38

38

## Accuracy

IMPERIAL

- Accuracy is a metric to evaluate classification models.
- It shows the fraction of predictions that the model has made correct predictions.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

39

39

## Accuracy – how to interpret it

IMPERIAL

- Accuracy could be a misleading metric without sufficient (other) complimentary information.
- Imagine we have a model that has classified 100 tumours as malignant (the positive class) and benign (the negative class).
- If someone tells you that the accuracy of the model is 91%; is that good?

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

40

40

## Accuracy interpretation example

IMPERIAL

- Of the 100 tumour examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 91 benign tumours, the model correctly identifies 90 as benign. That's good.
- However, of the 9 malignant tumours, the model only correctly identifies 1 as malignant—not a good outcome, as 8 out of 9 malignancies go undiagnosed!

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

41

41

## F<sub>1</sub> Score

IMPERIAL

- F<sub>1</sub> score is the harmonic mean of precision and recall.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Example: Calculate the F<sub>1</sub> score

True Positives: 25	False Positives: 10
False Negatives: 15	True Negatives: 80

42

42

## ROC curve (receiver operating characteristic curve)

IMPERIAL

- A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate (TPR)

- False Positive Rate (FPR)

$$TPR = \frac{TP}{TP + FN}$$

Does this look familiar?

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

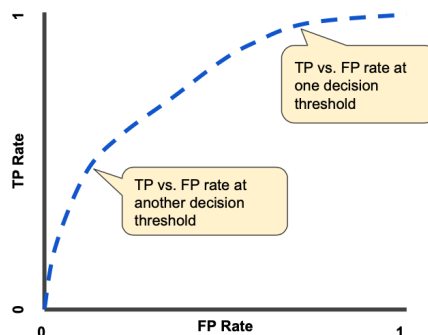
43

43

## ROC curve

IMPERIAL

- A ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.



source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

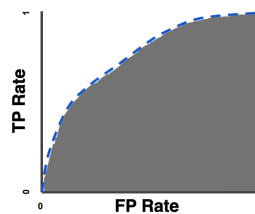
44

44

## AUC: Area Under the ROC Curve

IMPERIAL

- AUC stands for "Area under the ROC Curve."
- AUC measures the entire two-dimensional area underneath the entire ROC curve.
- AUC provides an aggregate measure of performance across all possible classification thresholds.



- Please note that there is also an AUC for Precision-Recall Curve

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

45

45

## PR AUC vs ROC AUC

IMPERIAL

### - PR AUC:

- Evaluating performance based on positive class
- Better suited for imbalanced data,
- Precision vs. Recall

### - ROC AUC:

- Evaluating overall classification performance,
- It can be misleading for imbalanced data,
- TPR vs. FPR

46

46

## Cases that AUC could be useful

IMPERIAL

- AUC is desirable for the following two reasons:
  - AUC is scale-invariant. It measures how well predictions are ranked rather than their absolute values.
  - AUC is classification-threshold invariant. It measures the quality of the model's predictions regardless of the classification threshold chosen.

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

47

47

## Cases that AUC will NOT be useful

IMPERIAL

- Scale invariance is not always desirable. For example, sometimes, we really do need well-calibrated probability outputs, and AUC won't tell us about that.
- Classification-threshold invariance is not always desirable. In cases where there are wide disparities in the cost of false negatives vs. false positives, it may be critical to minimise one type of classification error. For example, in a medical test that FPs and FNs do not carry equal weights. AUC isn't a useful metric for this type of optimisation.

source: <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>

48

48



## Prediction bias

IMPERIAL

- Some of the reasons:
  - The training set doesn't adequately represent certain subsets of the data space.
  - Some subsets of the data set are noisier than others.
  - The model is overly regularised.
- It can also be determined as:

$$\text{Prediction bias} = \text{average of predictions} - \text{average of labels}$$

49

49

## Sensitivity and specificity

IMPERIAL

- **Sensitivity:** the ability of a test to correctly identify patients with a disease.
- **Specificity:** the ability of a test to correctly identify people without the disease.
- **Prevalence:** the percentage of people in a population who have the condition of interest.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Source: Swift A, Heale R, Twycross A, What are sensitivity and specificity?, Evidence-Based Nursing 2020;23:2-4.

50

50

## Sensitivity/Specificity: Example

IMPERIAL

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Example: Calculate the sensitivity and specificity

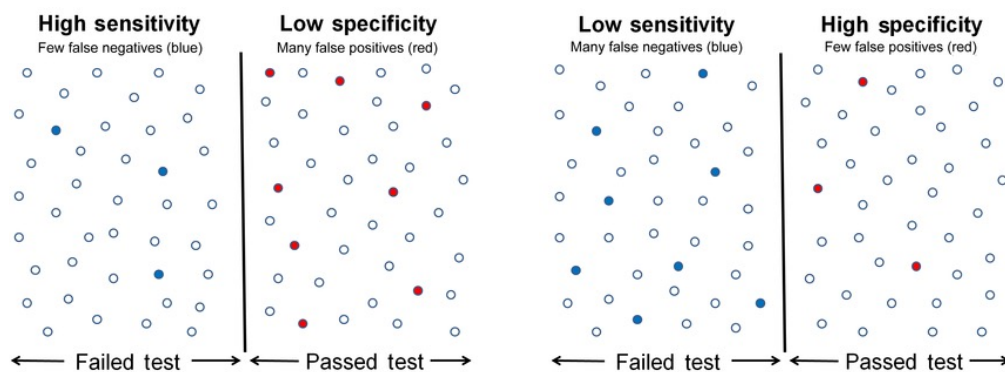
True Positives: 25	False Positives: 10
False Negatives: 15	True Negatives: 80

51

51

## Specificity and Sensitivity

IMPERIAL



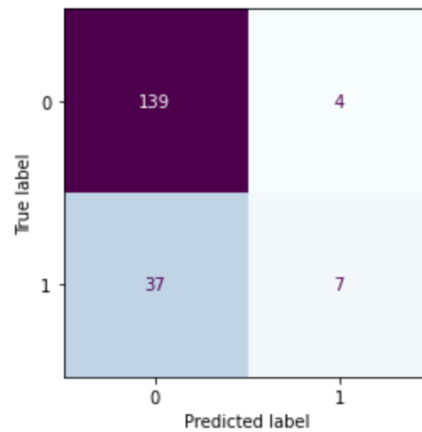
Source: [https://en.wikipedia.org/wiki/Sensitivity\\_and\\_specificity](https://en.wikipedia.org/wiki/Sensitivity_and_specificity)

52

52

## Confusion matrix

IMPERIAL



53

53

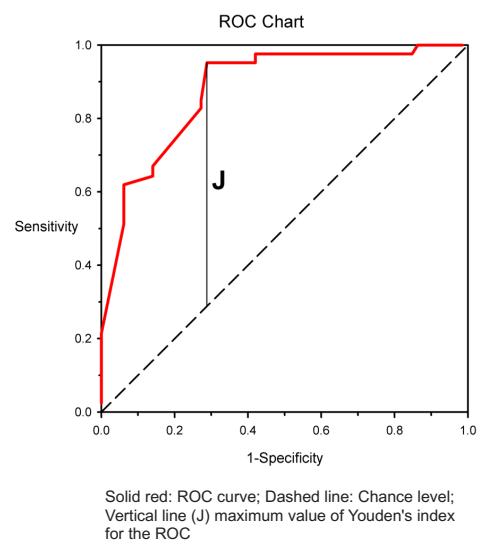
## Youden's index

IMPERIAL

- Captures the “informedness” measure..

$$J = \text{Sensitivity} + \text{Specificity} - 1$$

- The classification thresholds can be selected to maximise informedness.



Source: [https://en.wikipedia.org/wiki/Youden%27s\\_J\\_statistic](https://en.wikipedia.org/wiki/Youden%27s_J_statistic)

54

54

## Probability calibration

IMPERIAL

- Sometimes, in classification, you may want to predict the class label as well as the probability of the label for that class.
- This probability will provide an estimate/measure of the prediction.
- You need to be aware that some models can give poor estimates of the class probabilities (e.g. Naïve Bayes).
- Calibrating a classifier consists of fitting a regressor (called a calibrator) that maps the output of the classifier to a calibrated probability in  $[0, 1]$ .
- Denoting the output of the classifier for a given sample by  $f_i$ , the calibrator tries to predict  $p(y_i=1 | f_i)$ .
- See: <https://scikit-learn.org/stable/modules/calibration.html>

55

55

## All in one place

IMPERIAL

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

56

56

IMPERIAL

## Review Questions

57

57

IMPERIAL

Q1

- Imagine you come across a model for which its accuracy is specified as 91%; is that good?

*menti code will be provided.*

58

58

## Q2

IMPERIAL

- Imagine you are developing a screening tool for the risk of Alzheimer's disease using electronic healthcare records (EHR) data. You will implement an ML model to process historical medical records. Which metric will give you a higher weight (if any)?

59

59

## Metric in Scikit-learn

IMPERIAL

- Metrics and scoring:

[https://scikit-learn.org/stable/modules/model\\_evaluation.html#confusion-matrix](https://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix)

60

60

## Acknowledgments

**IMPERIAL**

- The slides on likelihood estimation are adapted from “A student’s guide to Bayesian Statistics) by Ben Lambert, SAGE, 2019.
- Some of the slides on Bayesian models and equations are adapted from Kevin Murphy’s book:
  - Machine Learning: A Probabilistic Perspective Kevin P. Murphy, MIT Press.

61

61

## If you have any questions

**IMPERIAL**

- Please feel free to arrange a meeting or email ([p.barnaghi@imperial.ac.uk](mailto:p.barnaghi@imperial.ac.uk)).
- My office: 928, Sir Michael Uren Research Hub, White City Campus.

62

62