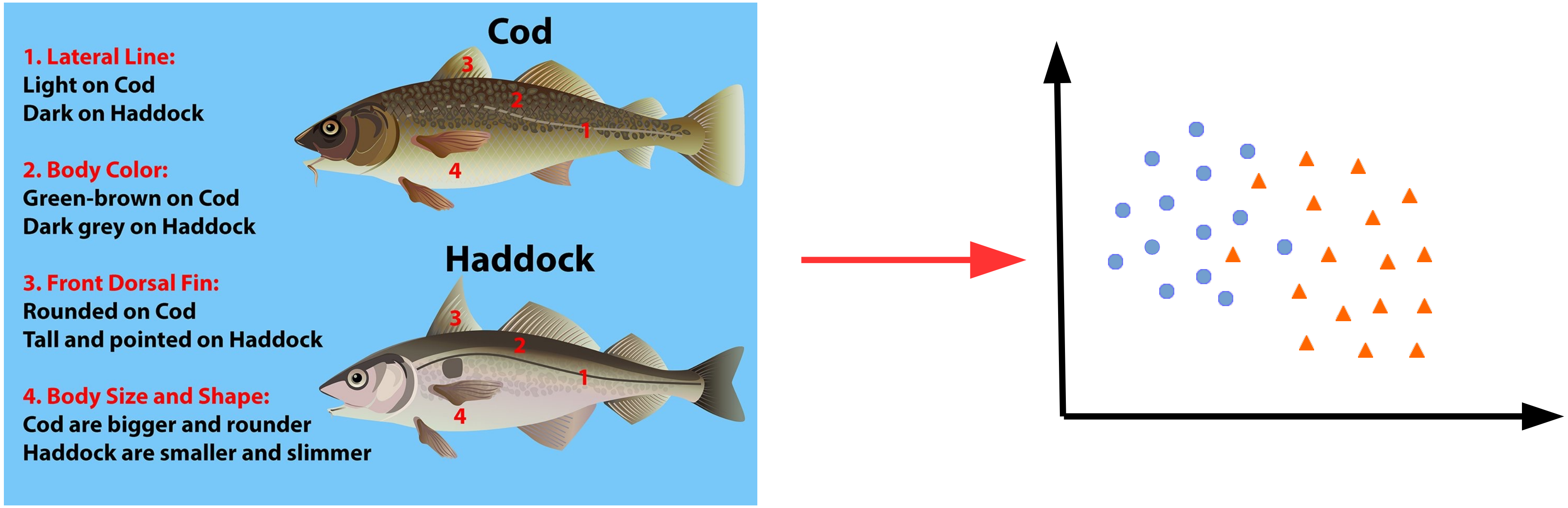
UNIVERSITY OF LEEDS

Machine Learning Evaluation

Learning outcomes

- Define generalization.
- Define overfitting.
- Apply a strategy to avoid overfitting.
- List the main accuracy metrics to measure the performance of a classifier.
- Choose the appropriate metric for a given classification problem.
- Apply the metrics to real data sets and classifiers.

Feature Selection



1. Lateral Line:
Light on Cod
Dark on Haddock

2. Body Color:
Green-brown on Cod
Dark grey on Haddock

3. Front Dorsal Fin:
Rounded on Cod
Tall and pointed on Haddock

4. Body Size and Shape:
Cod are bigger and rounder
Haddock are smaller and slimmer

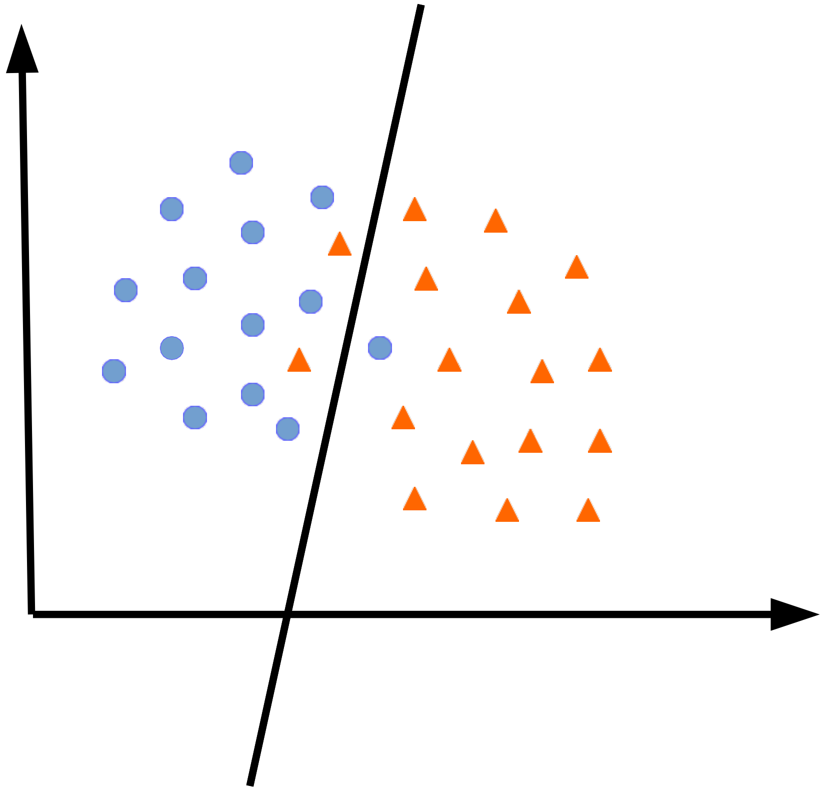
The first step that can take place before classification is to decide what *features* we are considering when trying to discriminate two sets.

For example, we could measure width, height, colour, and/or shape…

Example: parametric classifier

Most classifiers have parameters to tune, for instance:

We are looking for a straight line to separate the data. Parameters: $y = \boldsymbol{m}\,x + \boldsymbol{q}$



Training versus test data

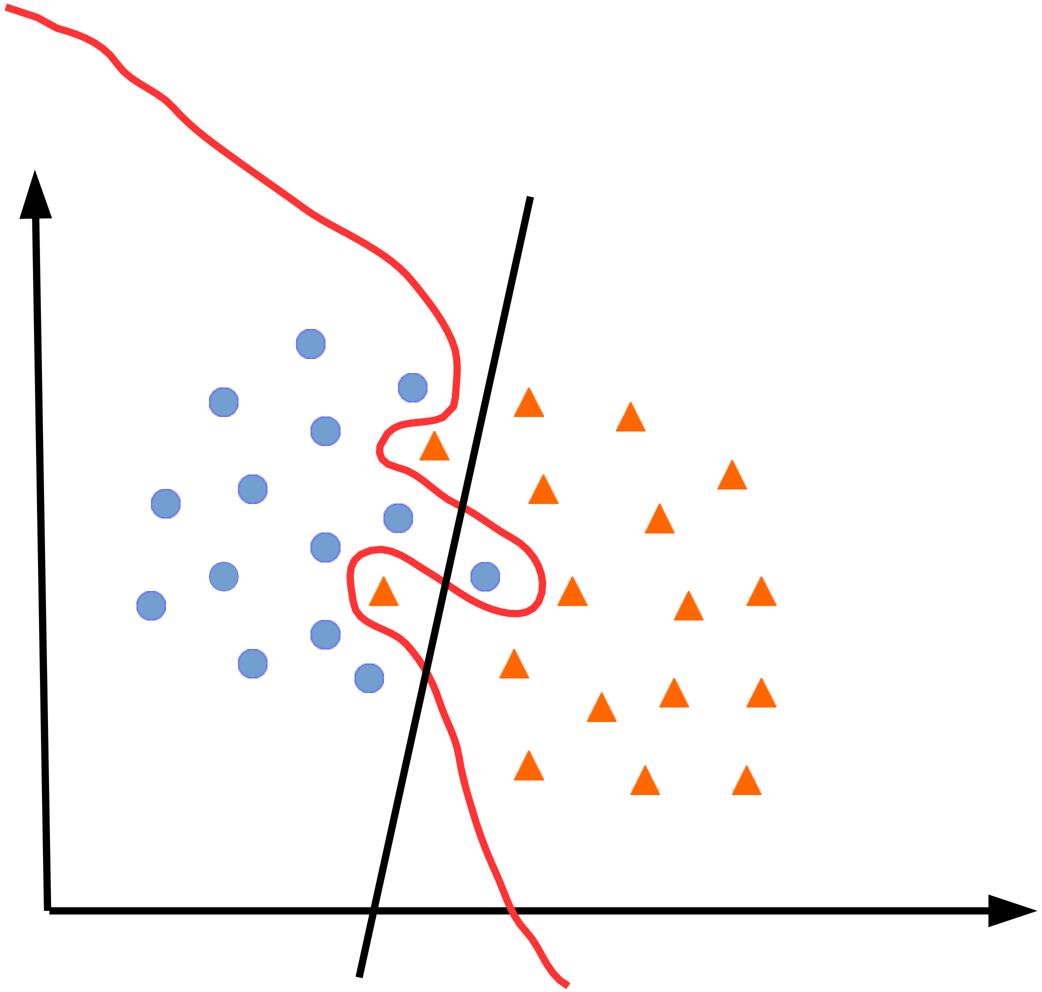
How can we correctly evaluate the performance of the model? Test on a portion of the data different from training.

Training data

Test data

Generalization:
The ability of the classifier to correctly classify an unseen data

Model complexity



Which one would you say it's best?

Overfitting vs underfitting

- A model *overfits* when it describes the randomness associated with the data, rather than the underlying relationship between the data points.

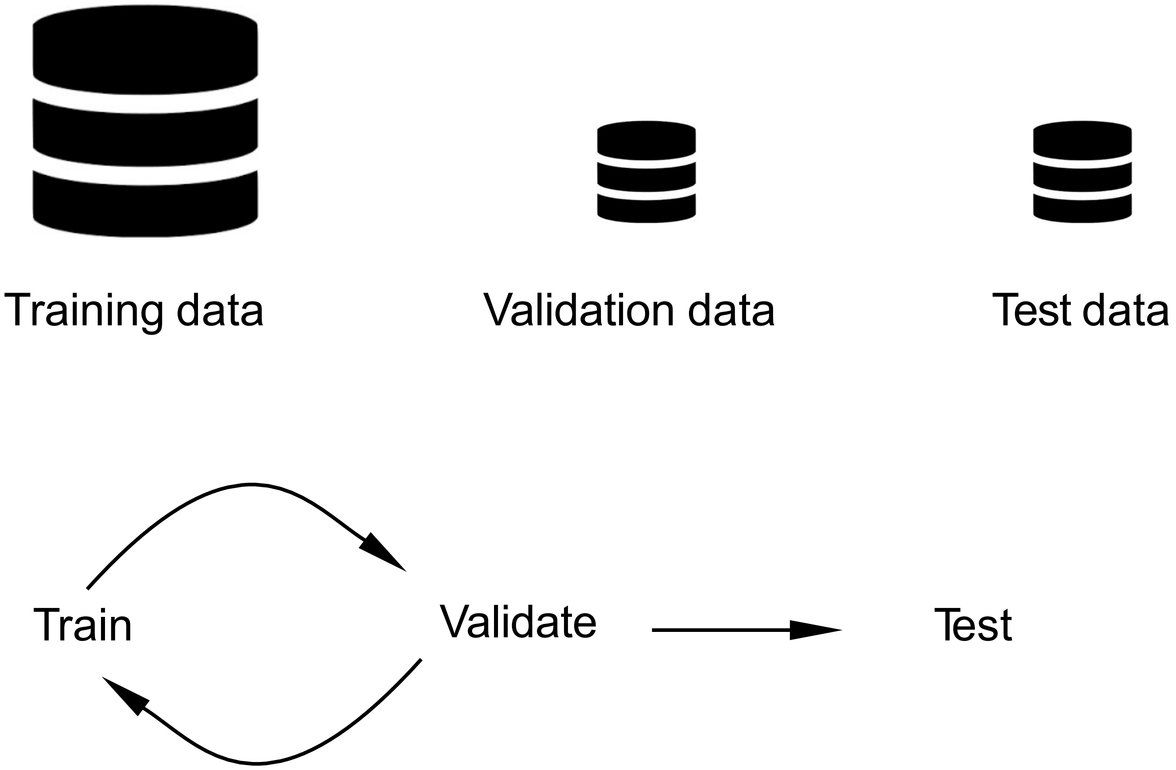
- Occam’s razor principle: Do not introduce complexity if not necessary

- A model underfits when it fails to capture the true complexity of the data distribution.

- A good balance between the complexity of the model and amount of the data must be established.

Preventing overfitting

Validafion set



Training data

Validation data

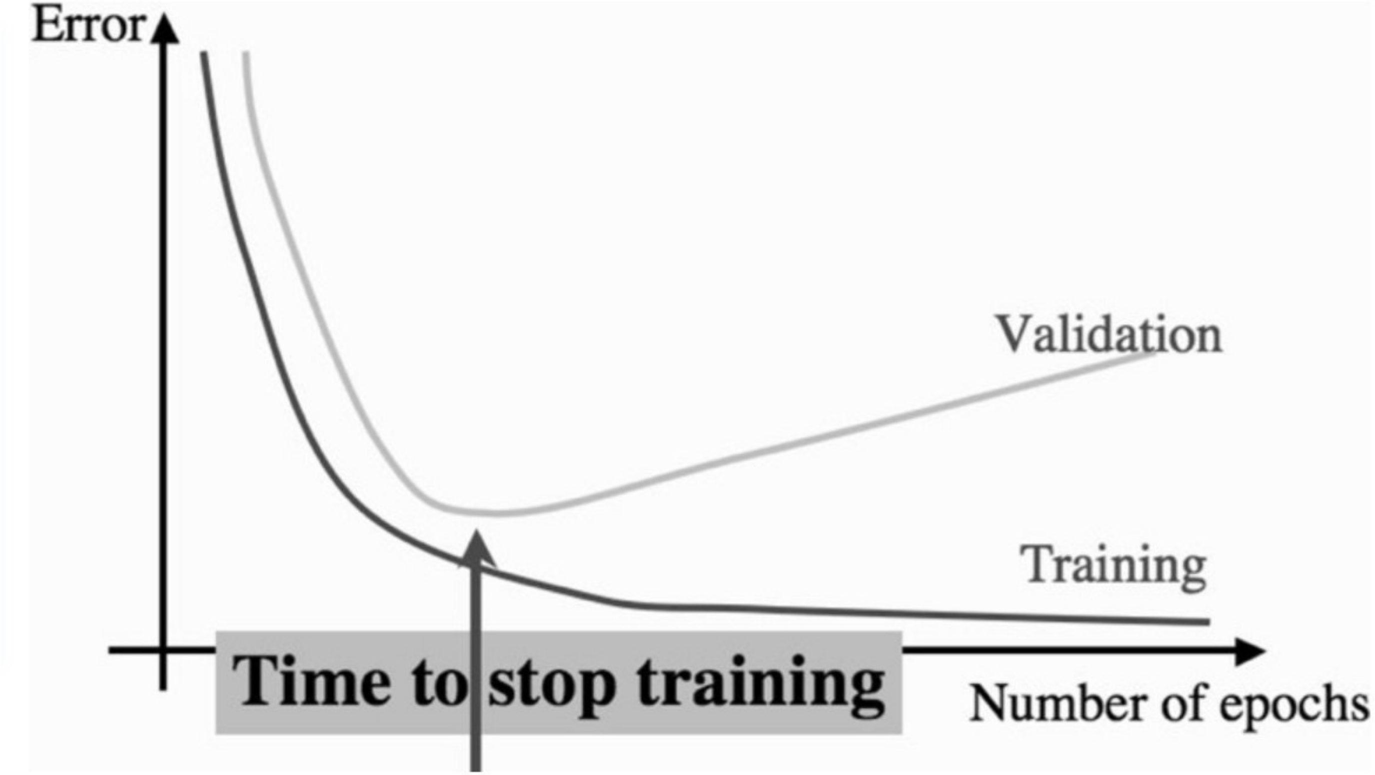
Test data

Train

Validate

Test

When to stop learning



Question

The accuracy on the training set is 90%. Should I add parameters to the model and aim at making it 100%?

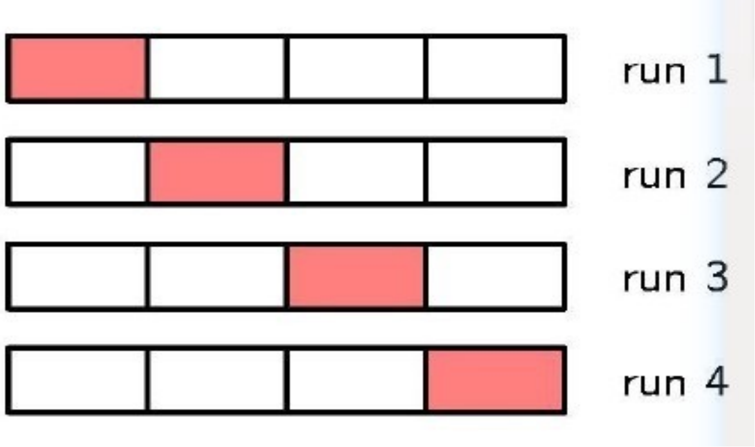
1 - Yes, the higher the accuracy, the better.

2- No, the lower the accuracy on the training data, the higher the generalisation

3- I should test the accuracy of both models (original, and with more parameters) on a test set, and only then decide which one is best.

S-fold cross validation

- S fold cross validation involves partitioning it into S groups.
- In each run, S − 1 groups are used to train the model and validation error is evaluated using the held-out group (red).
- This procedure is then repeated for all S possible choices for the held-out group, and the performance scores from the S runs are then averaged.
- At the end, each data point has contributed both to validation and training.



Question

What is the validation set for?

1- If we have too much data, we can get rid of some in the validation set.

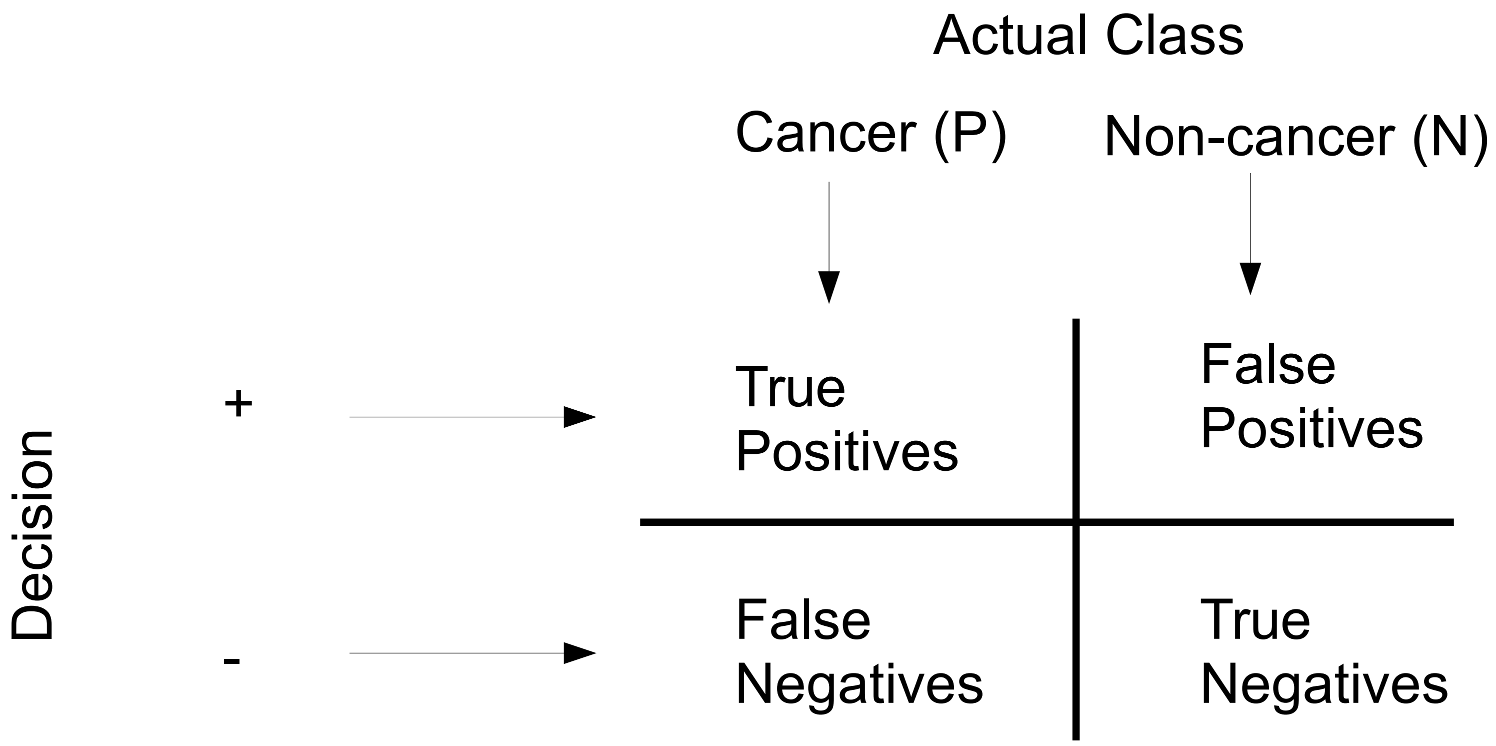
2- We use the validation set to check for overtitting while we are still optimising it, but then for testing we need a separate test set.

3- We can test our models on both test and validation, for more accurate results.

Performance measures

Confusion matrix of a binary classitier

Suppose have a binary classifier that uses a blood test to detect cancer as class P versus healthy as class N.

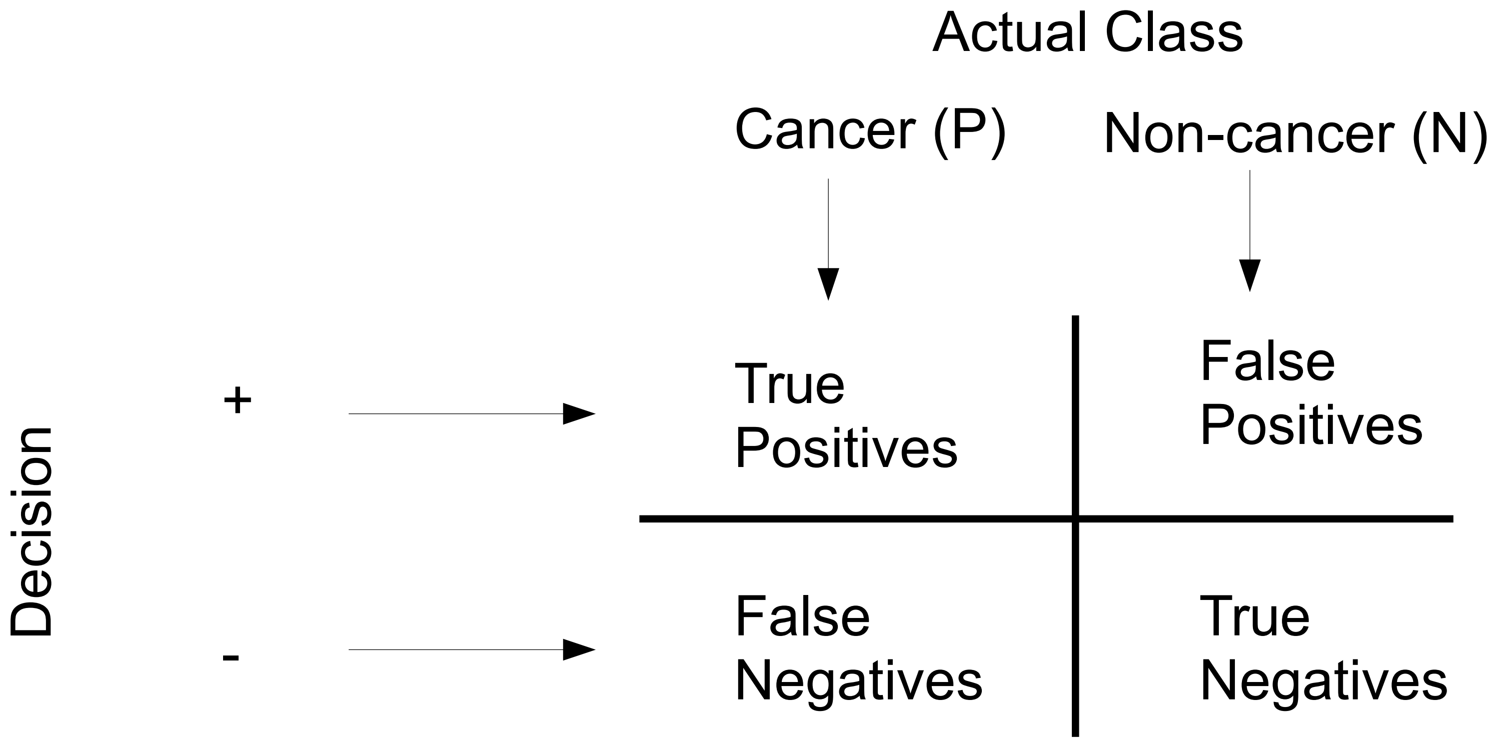


Actual Class

Decision	Cancer (P)	Non-cancer (N)
+	True Positives	False Positives
-	False Negatives	True Negatives

Confusion matrix of a binary classitier

Suppose have a binary classifier that uses a blood test to detect cancer as class P versus healthy as class N.



Actual Class

Decision	Cancer (P)	Non-cancer (N)
+	True Positives	False Positives
-	False Negatives	True Negatives

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

Confusion matrix of a binary classitier

Suppose have a binary classifier that uses a blood test to detect cancer as class P versus healthy as class N.

		Actual Class	
		Cancer (P)	Non-cancer (N)
Decision	+	True Positives	False Positives
	-	False Negatives	True Negatives

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

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

Confusion matrix of a binary classitier

UNIVERSITY OF LEEDS

Suppose have a binary classifier that uses a blood test to detect cancer as class P versus healthy as class N.

		Actual Class	
		Cancer (P)	Non-cancer (N)
Decision	+	True Positives	False Positives
	-	False Negatives	True Negatives

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

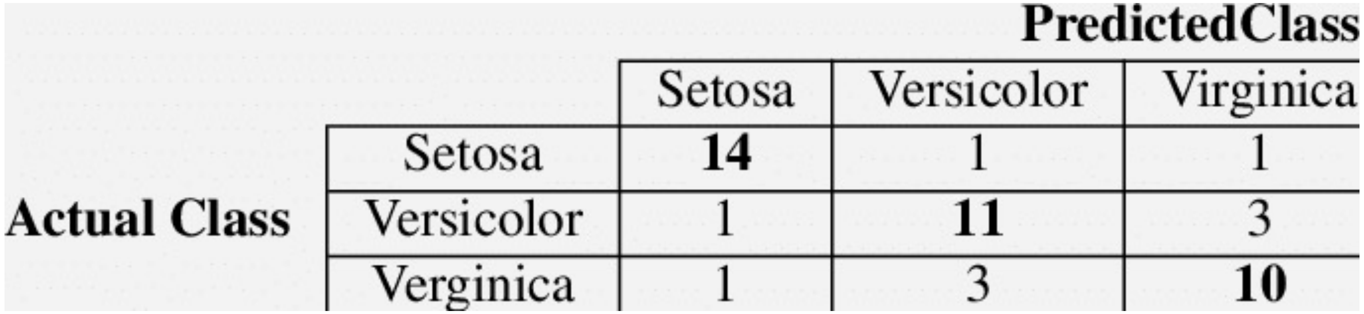
$$\text{Recall} = \frac{TN}{TN+FN}$$

Confusion Matrix

- It is convenient to generalise the confusion matrix to evaluate the accuracy of multi-class classifiers.

- Each entry at coordinate (i, j) in the matrix corresponds to the number of elements of class i samples classified as j.

- For instance, a classifier could result in the following confusion matrix when applied to Iris flower data:

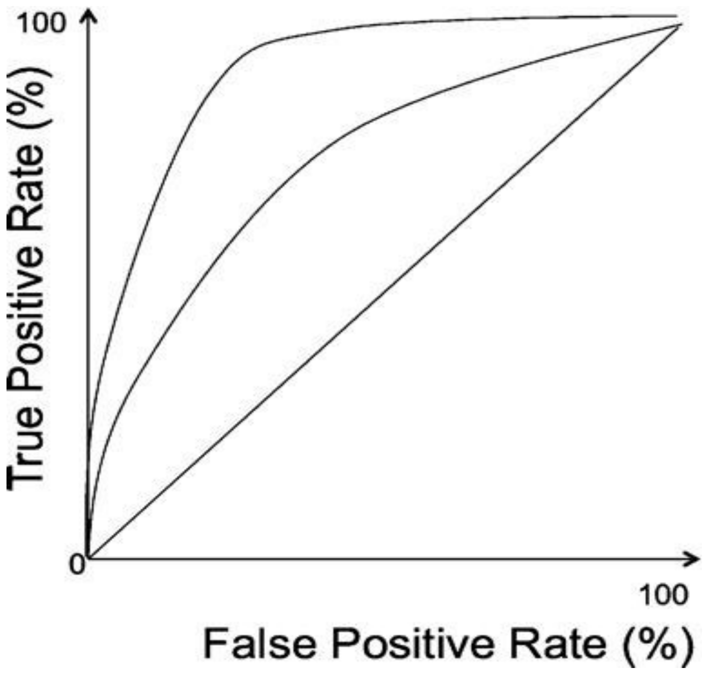


		PredictedClass		
		Setosa	Versicolor	Virginica
Actual Class	Setosa	**14**	1	1
	Versicolor	1	**11**	3
	Verginica	1	3	**10**

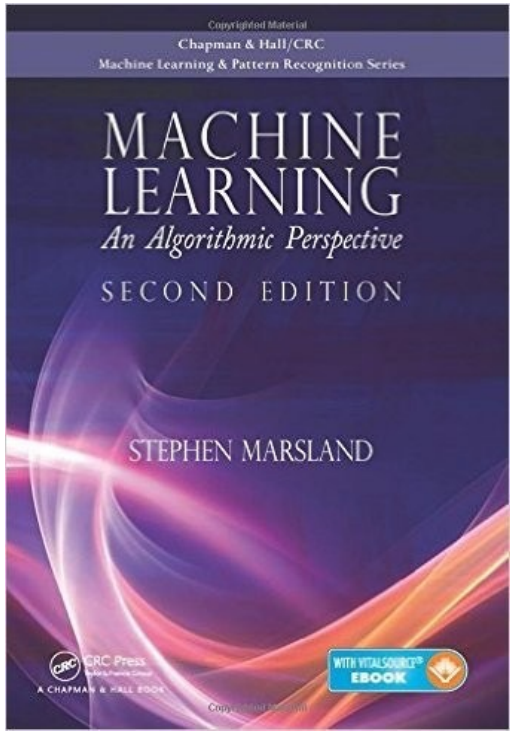
Receiver Operator Curve (ROC)

UNIVERSITY OF LEEDS

- Binary classifiers generally generate a continuous likelihood value for one class versus the other.
- A discrimination threshold must be selected and applied to this likelihood value to decide the classes.
- The ROC curve shows how TP and FP vary as the discrimination threshold varies.



- The area under ROC curve is often utilized to compare different classifiers, regardless of the threshold value.



Chapter 2, up to 2.2