Machine Learning

Evaluation

Dr Jian Liu

Machine Learning

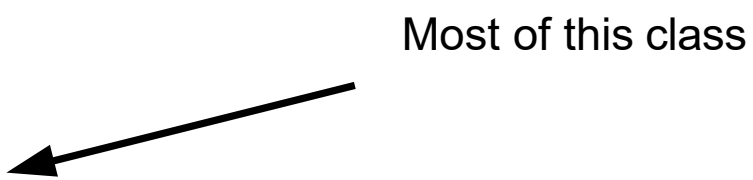
Data Collection

Feature Selection

Model choice

Training

Evaluation



Evaluation

- 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

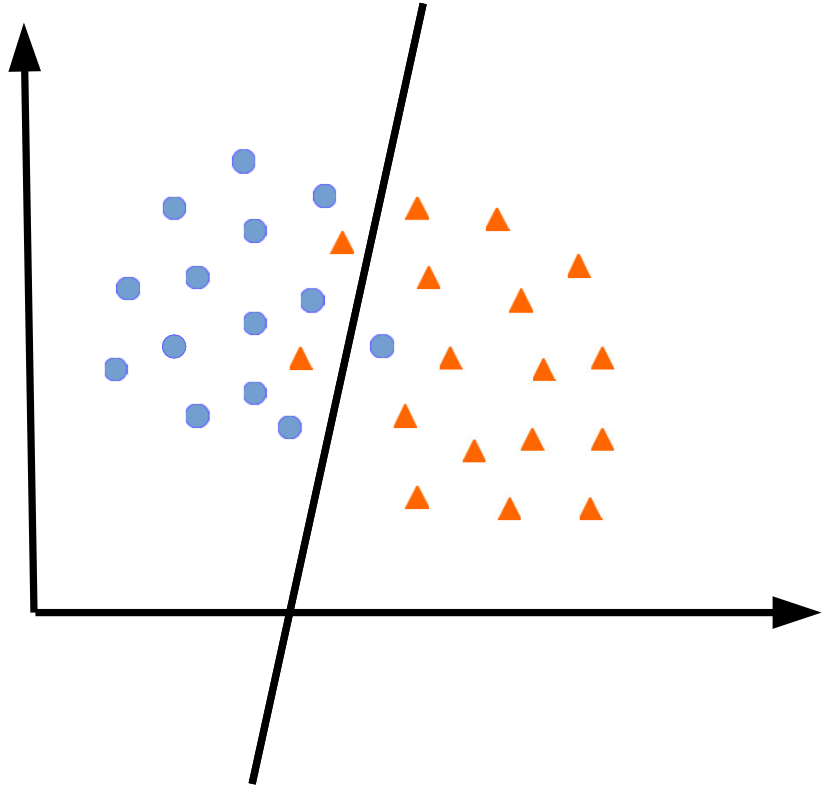
The first step that can take place before classification is to decide what *features* we are considering when trying to discriminate two sets.

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

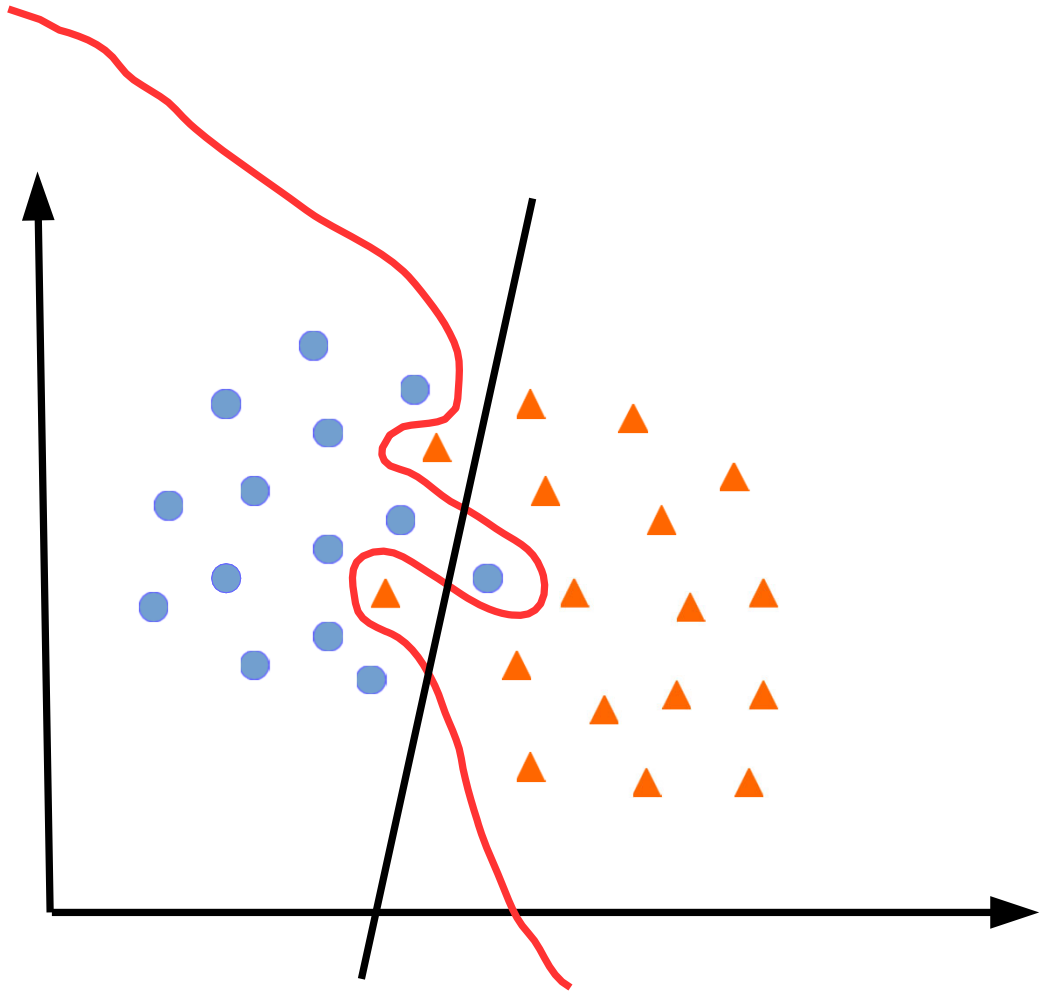
How can we correctly evaluate the performance of the model? Test on a portion of the data different from training.

Training data

Test data

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

Model complexity



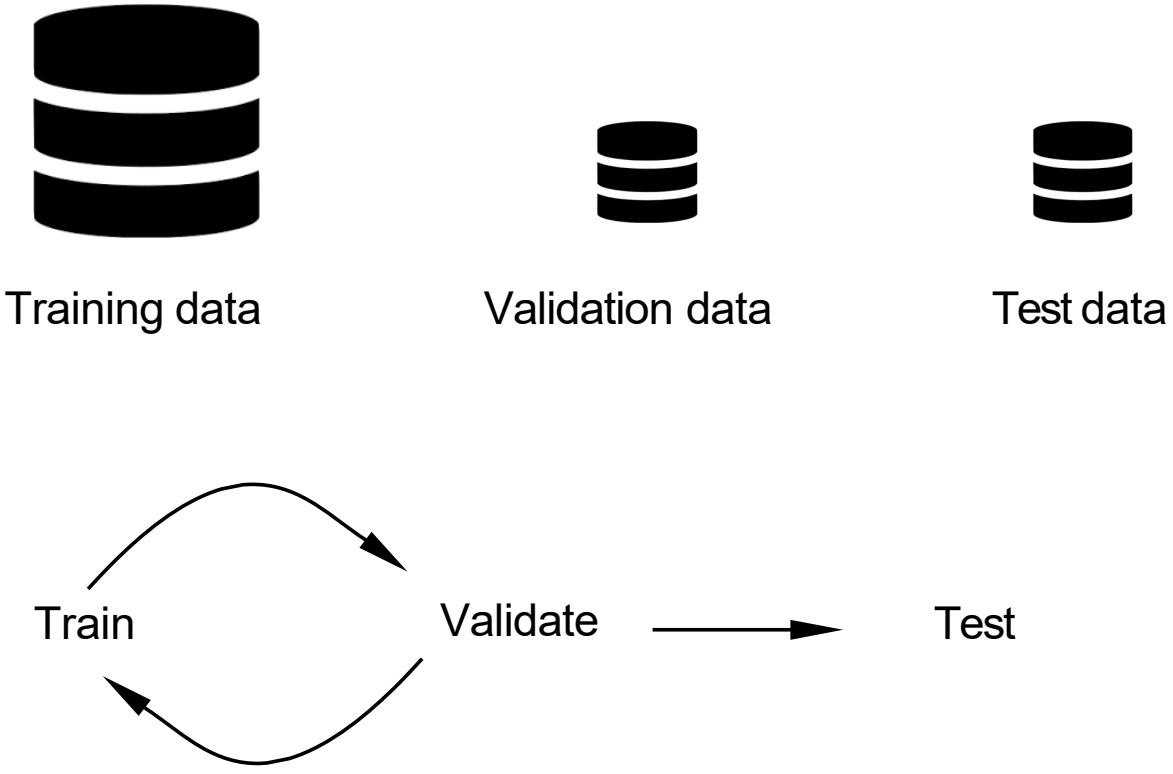
Which one would you say it's best?

Overfitting vs underfitting

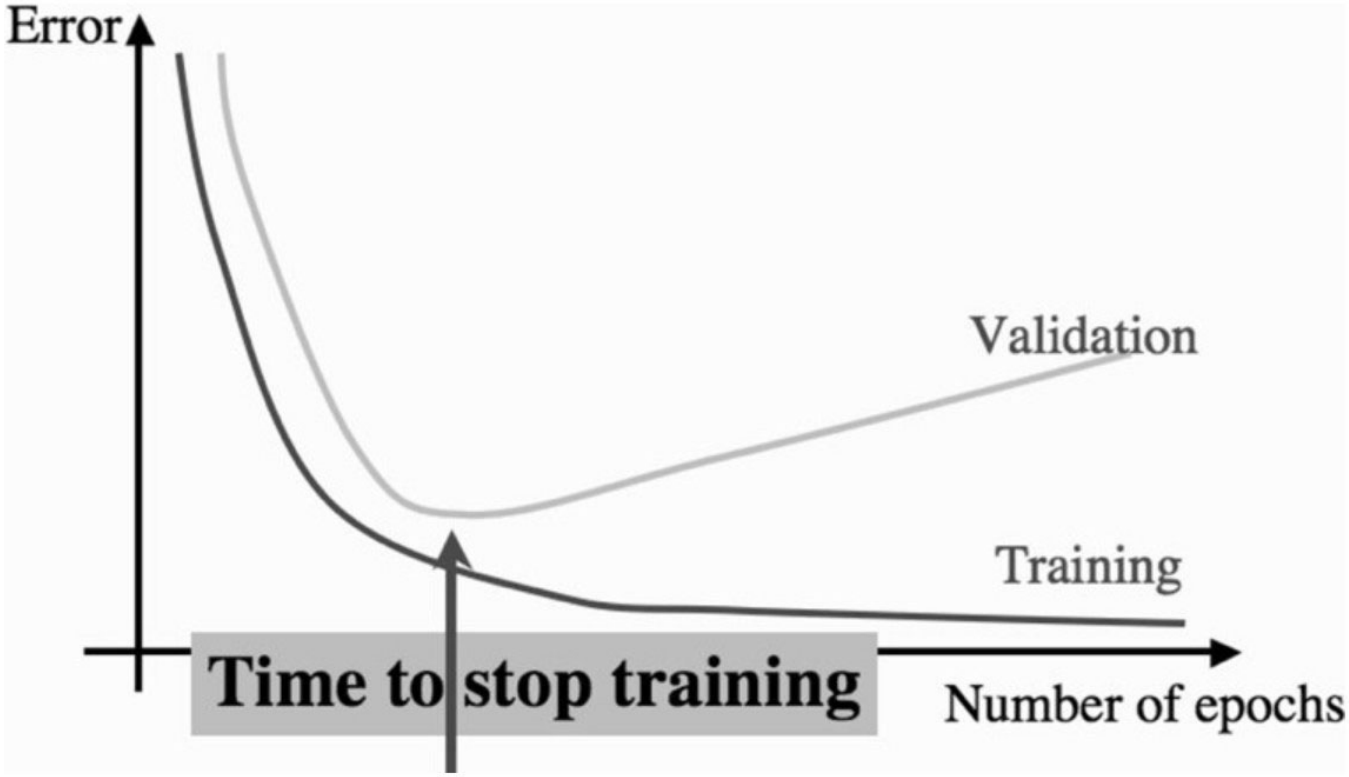
- A model *overfits* when it describes the randomness associated with the data, rather than the underlying relationship between the data points.
- Occam's razor principle: Do not introduce complexity if not necessary (aka the KISS principle)
- A model underfits when it fails to capture the true complexity of the data distribution.
- A good balance between the complexity of the model and amount of the data must be established.

Preventing overfitting

Validation set



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%?

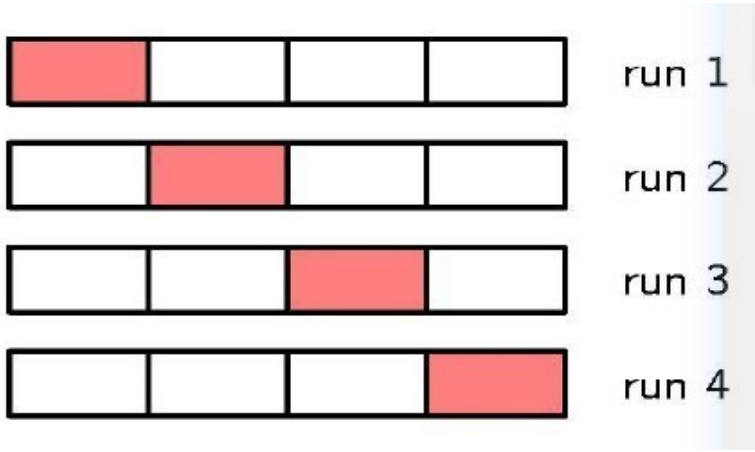
1 - Yes, the higher the accuracy, the better.

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

3 - One 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

- Sfold cross validation involves partitioning it into Sgroups.
- 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.

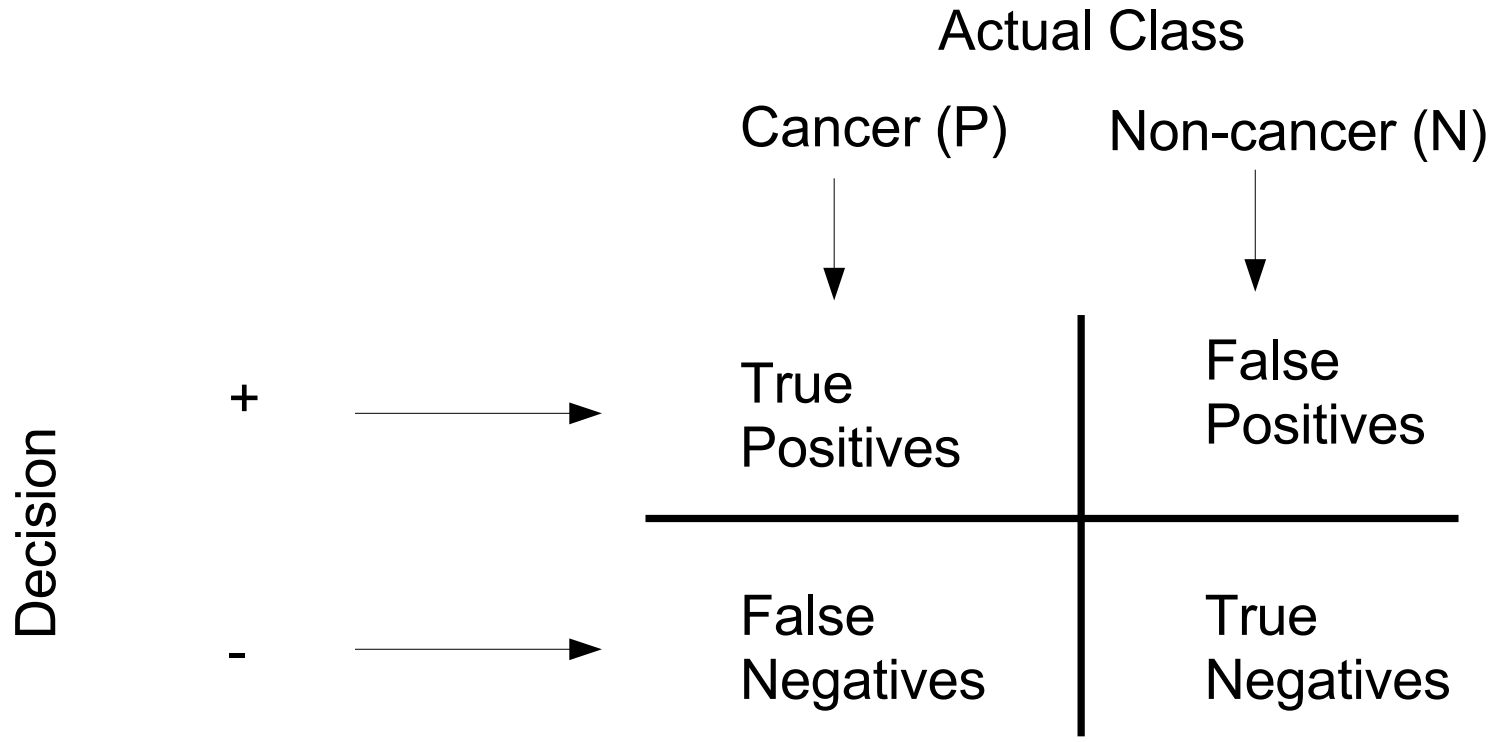
3 - We can test our models on both test and validation, for more accurate results.

Performance measures

Confusion matrix of a binary classifier

UNIVERSITY OF LEEDS

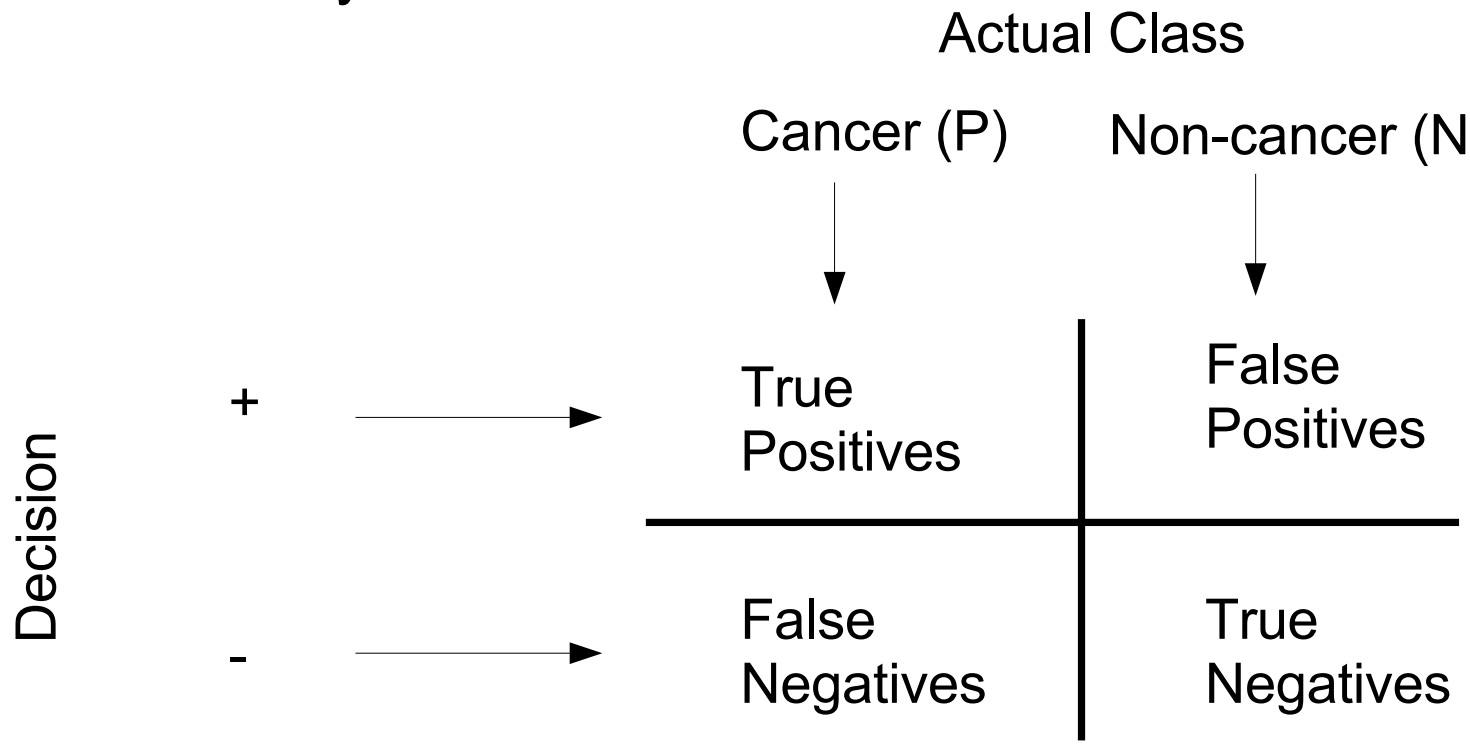
Suppose have a binary classifier that uses a blood test to detect cancer as class P versus healthy as class N.



Confusion matrix of a binary classifier

UNIVERSITY OF LEEDS

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



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

Confusion matrix of a binary classifier

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

What proportion of positive identifications was actually correct?

Sensitivity / Recall / True Positive Rate

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

What proportion of the positive class got correctly classified.

A simple example would be to determine what proportion of the actual sick people were correctly detected by the model.

Receiver Operator Curve (ROC)

Er VSG $gyvzi ,vigimziv$stivexmrk$
glevegxivmwxmg$gyvzi-$w$e$kvetl$wls{mrk$
xli$tivjsvq ergi$sj$e$gpewwmjmexmsr$
q shipexepp$gpewwmjmexmsr$xlviwlsphw2

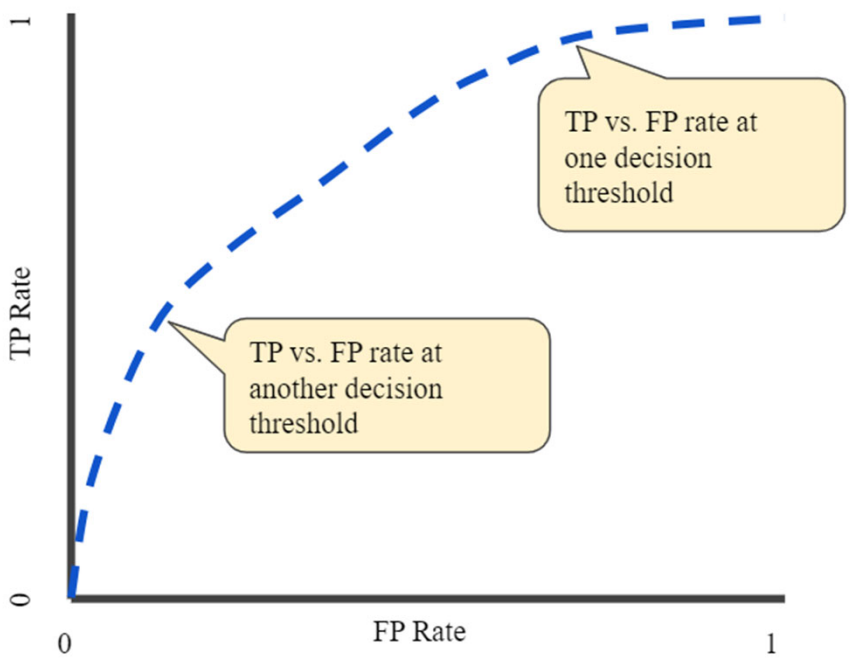
Xlmw$gyvzi$tsmxw$x{ s$teveq ixiw>

Xvyi$Tswmxmzi$Vexi ,XTV-we$w}rsr}q jsvvigep$
erh$w$xlivijsvi$hijmrih$ew$jspps{ w>

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

Jepwi$Tswmxmzi$Vexi ,JTV-$w$hijmrihewjspps{ w>

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



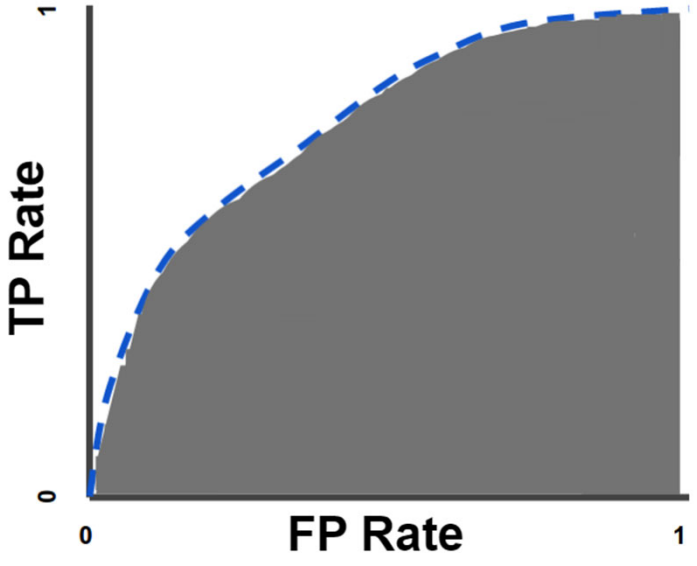
ErVSGgyvzi$tsmxw$XTV$zw2$JTV$
ex$hmjjivirx$gpewwmjmexmsr$
xlviwlsphw2

AUC: Area Under the ROC Curve

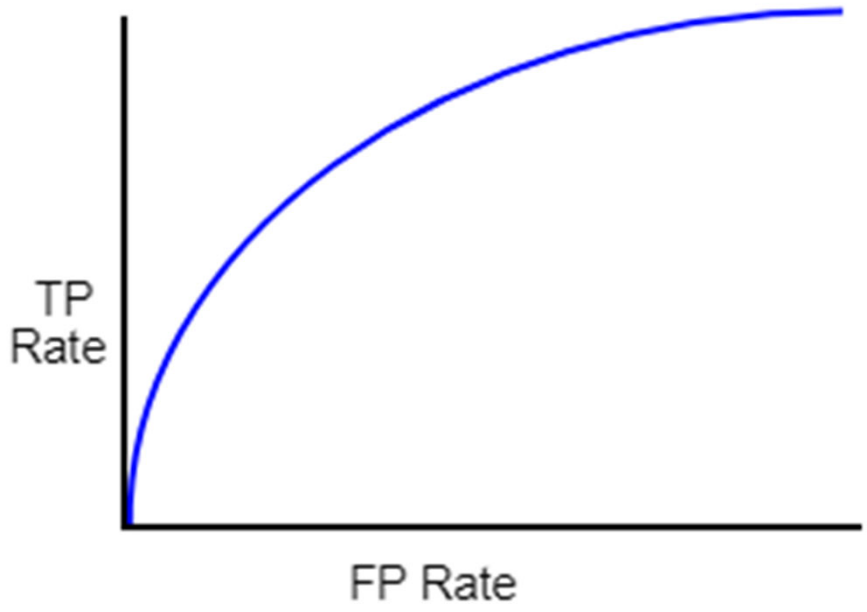
UNIVERSITY OF LEEDS

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

AUC is often utilized to compare different classifiers, regardless of the threshold value.

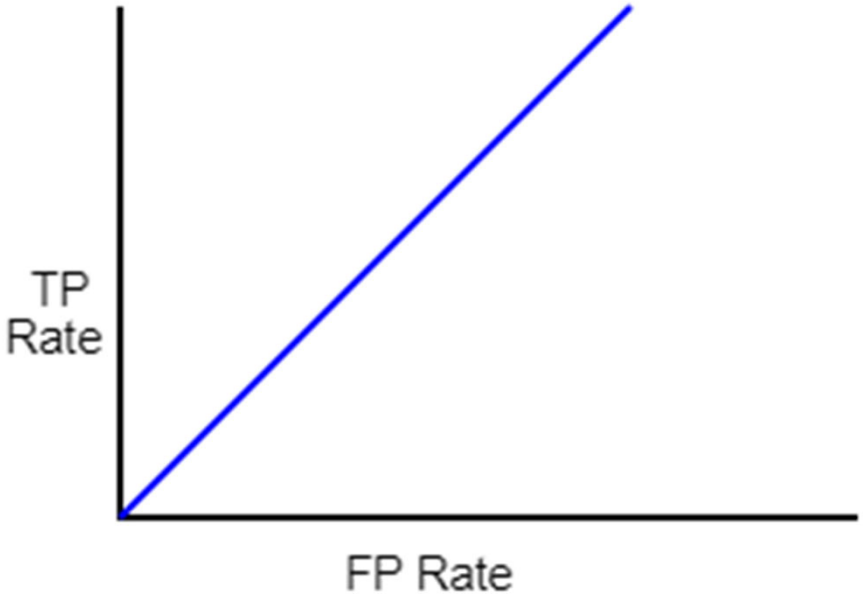


ROC and AUC



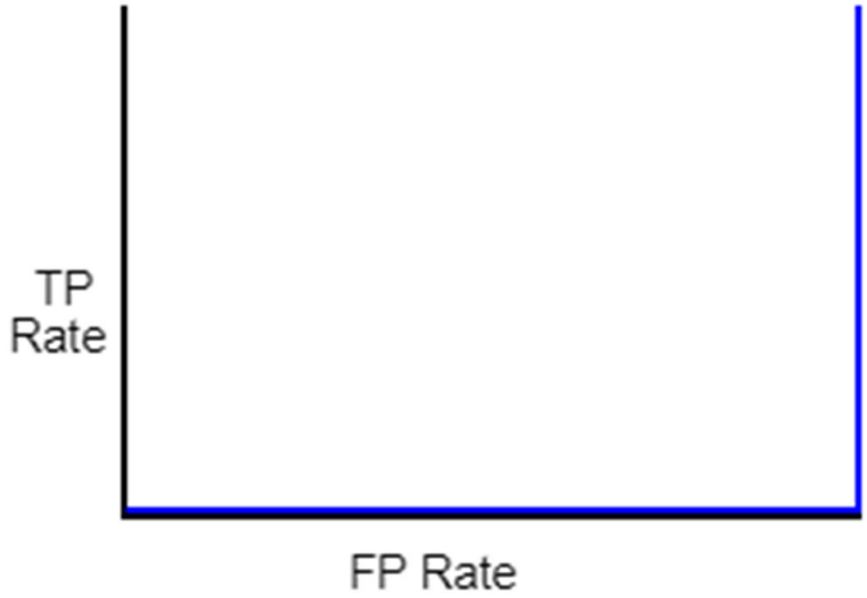
This ROC curve has an AUC between 0.5 and 1.0, meaning it ranks a random positive example higher than a random negative example more than 50% of the time. Real-world binary classification AUC values generally fall into this range.

ROC and AUC



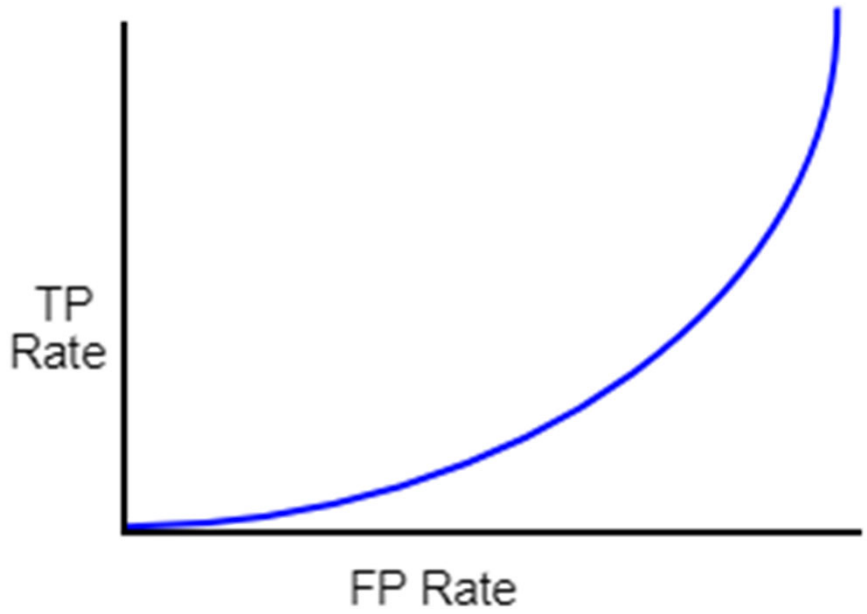
This ROC curve has an AUC of 0.5, meaning it ranks a random positive example higher than a random negative example 50% of the time. As such, the corresponding classification model is basically worthless, as its predictive ability is no better than random guessing.

ROC and AUC



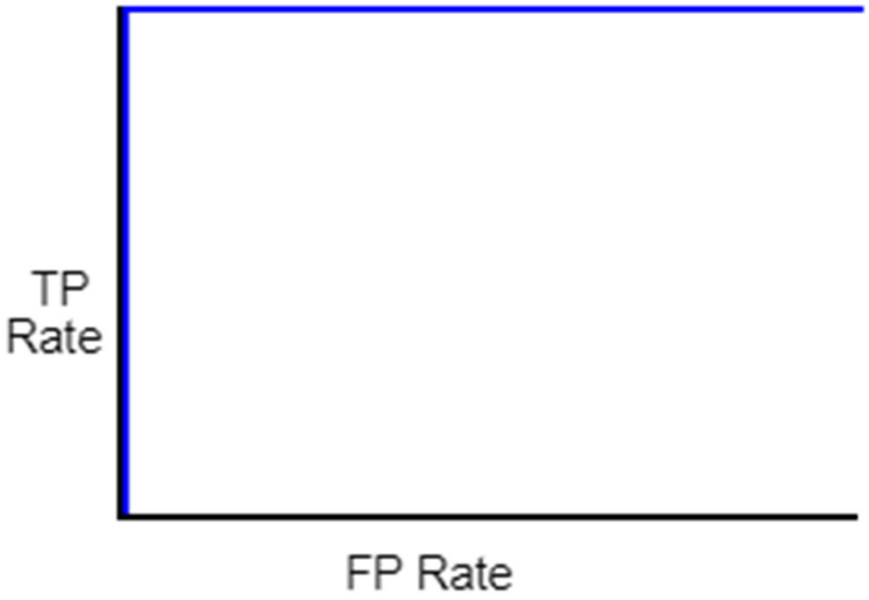
This is the worst possible ROC curve; it ranks all negatives above all positives, and has an AUC of 0.0. If you were to reverse every prediction (flip negatives to positives and postives to negatives), you'd actually have a perfect classifier!

ROC and AUC



This ROC curve has an AUC between 0 and 0.5, meaning it ranks a random positive example higher than a random negative example less than 50% of the time. The corresponding model actually performs worse than random guessing! If you see an ROC curve like this, it likely indicates there's a bug in your data.

ROC and AUC



This is the best possible ROC curve, as it ranks all positives above all negatives. It has an AUC of 1.0.

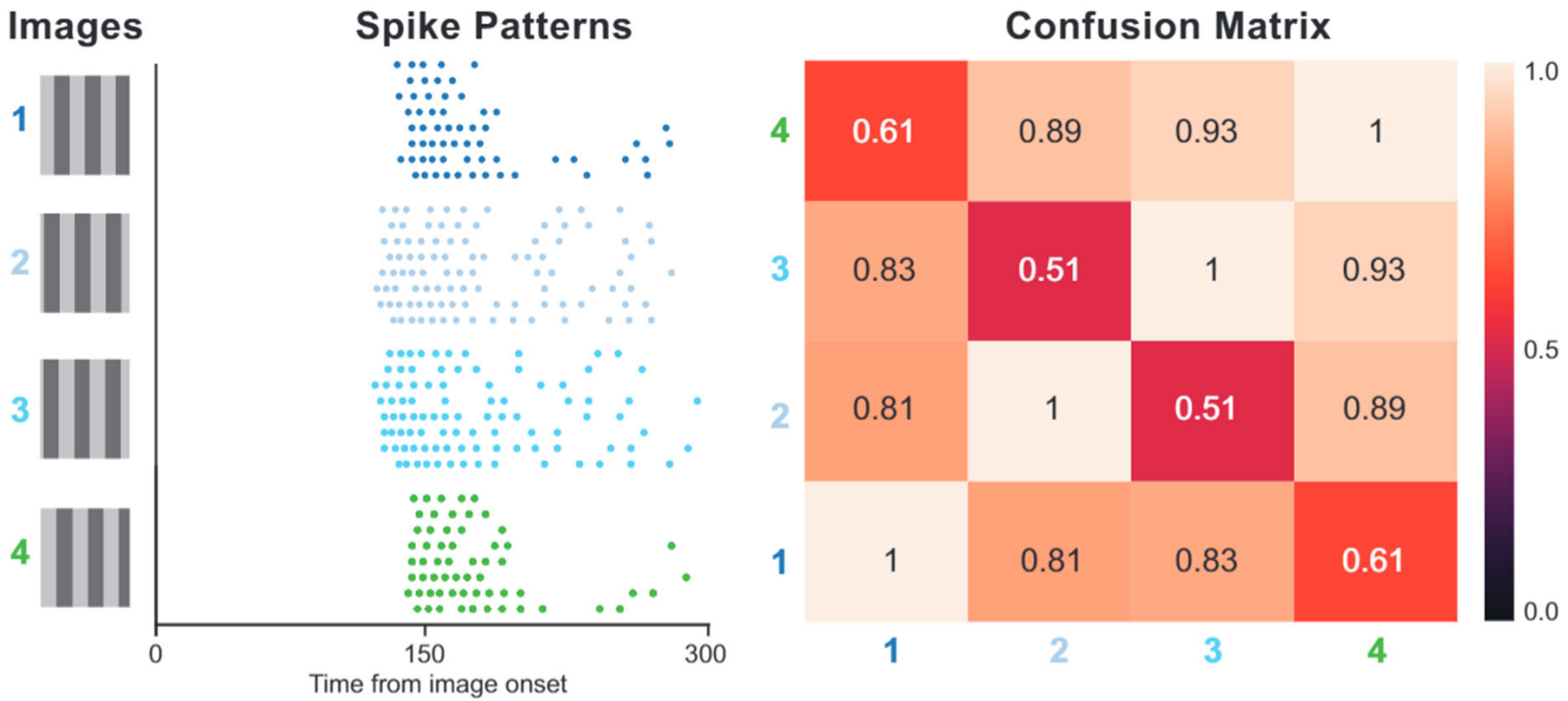
Mr$tvegxmgi$hj$}sy$lezi$e$tivjigx$gpewwmjmiv${ mxl$erEYG$sj524}sy$wlsyph$fi$wywtmgmsyw$
ew$xsqoip}$rhmgexiw$e$fyk$r$}syv$q shipx$Jsv$i|eq tpi$}sy$q e}$lezi$zivjmx$s$}syv$
xvemrmrk$hexe$svsxliefip$hexe$q e}fivitpgexih$r$risj}syv$jiexyviw2

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**

Confusion Matrix



Four patterns of Brain signals