COMP309 in Week 08, 2024

measuring performance (part 1)

Marcus Frean marcus@ecs.vuw.ac.nz

this week

Concepts of Performance Metrics

- Performance metrics vs loss function

Various classification Metrics

- Accuracy, true +ve/-ve rates, etc.
- ROC curve and AUC

Various Regression Metrics

- MSE, RMSE, RSE, MAE

Various Clustering Metrics

- Silhouette Score, Rand Index, Mutual Information

A concept: “log loss” as a metric, if learner outputs probabilities.
Revision of train / test / validation / cross-validation “gotchas”

today

next lecture

Performance Metrics - what’s the point

1. Metrics are ways to quantify / report an algorithm’s performance. E.g.,
   - a classifier, putting labels on data: we could report the accuracy (?)
   - a robot trained to navigate with Reinforcement Learning: a robot navigation task: total rewards (?)
   - prediction or interpolation of reals using a regressor : mean squared error (?)

2. Helpful to decide the best model to meet the target performance.
   - E.g., for a simple classification task, you have a pool of algorithms to choose from, such as kNN, Naïve-bayes, DT, etc.

     Use a metric to decide which one to use.

Performance Metrics *versus* Loss Functions

Performance Metrics are objectives that we wish we could optimise (i.e., functions used for evaluating the quality of the models)

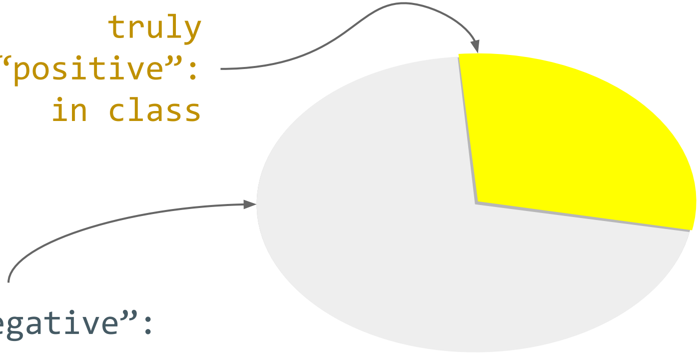
- Example: we don't *really* care about mean-squared error for regression: it's really about whether our interceptor hits the next missile…
- Concept: we want to improve accuracy, on unseen data (at least, in classification tasks), but it is difficult to directly optimise that efficiently.

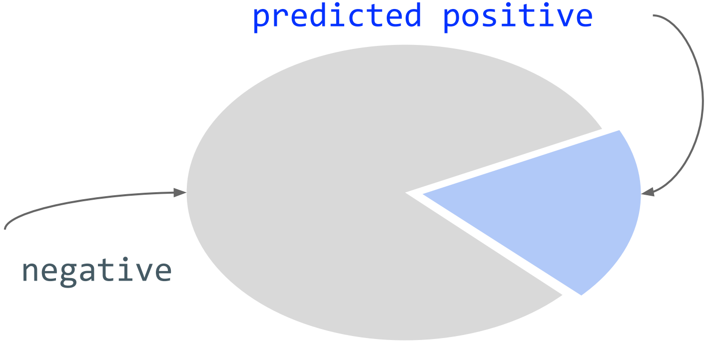
Loss functions are functions used as proxy metrics to be actually optimised

- Example: for regression tasks, the mean squared error on the train data is often used.
- Example: for classification, the log loss is usually used.

(aside: note the words *Loss*, *log Loss*, *Cost function*, *Objective*, *Fitness*, *Utility* are used / confused as synonyms)

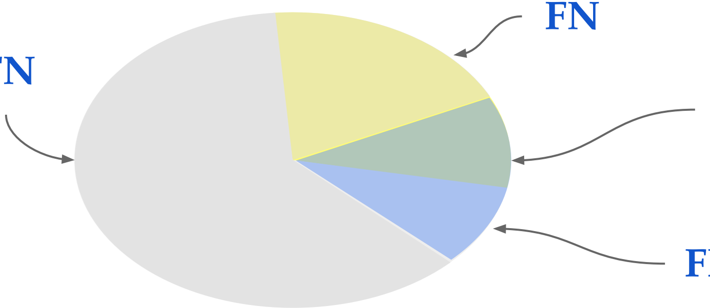
Performance Metrics for binary classification





		ground truth	
		P	N
predicted	P	**TP**	**FP**
	N	**FN**	**TN**

aptly known as a “Confusion Matrix” 😕



Performance Metrics

The following are somewhat “blunt” instruments – do you see why?

Accuracy: (TP+TN)/(TP+TN+PF+FN)

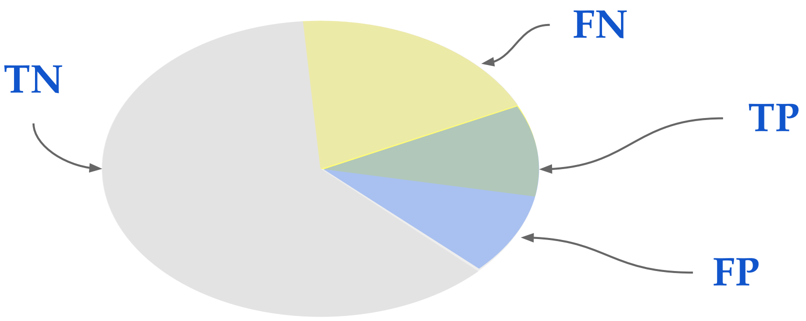
- how often is the classifier correct?

Misclassification (Error) Rate, (1- Accuracy)

- how often is it wrong?

		ground truth	
		P	N
predicted	P	TP	FP
	N	FN	TN

aptly known as a “Confusion Matrix” 😕



Performance Metrics for binary classification

True Positive Rate, or Sensitivity, or Recall:

TP/(TP+FN)

how often does predictor say "positive" <u>on cases that are *actually positive*</u>? (also known as "true positive rate (TPR)", or "recall"). Biggest if says "yes" to almost anything ⇒ a highly sensitive classifier might be 'reckless'

False Positive Rate (FPR):

FP/(TN+FP)

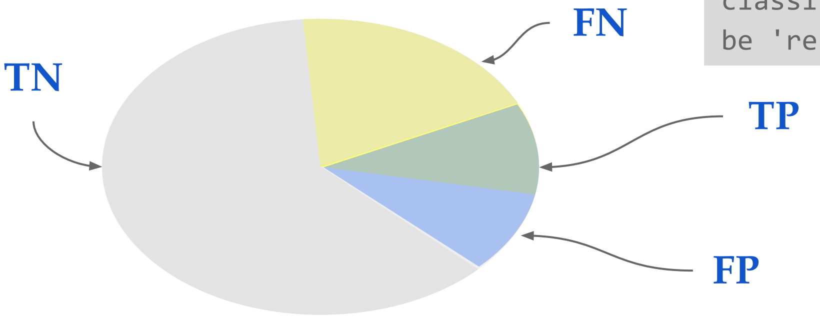
how often does predictor say "positive" <u>on cases that are in fact *negative*</u>?

Specificity:

TN/ (TN+FP)

how often does predictor say "negative" <u>on cases that are in fact negative?</u>
Also known as true negative rate (TNR), (note it's equivalent to 1 – FPR)

		ground truth	
		P	N
predicted	P	TP	FP
	N	FN	TN



Performance Metrics

c.f. Recall: TP/(TP+FN)

Precision: TP/(TP+FP)

When your classifier predicts positive, how often is the case *actually* positive?

Easiest way to have high precision is predict "yes" for only strongly +ve cases ⇒ a high precision classifier might be 'conservative'

how often does predictor say "positive" on cases that are *actually positive*? (also known as "true positive rate (TPR)", or "recall"). Biggest if says "yes" to almost anything ⇒ a highly sensitive classifier might be 'reckless'

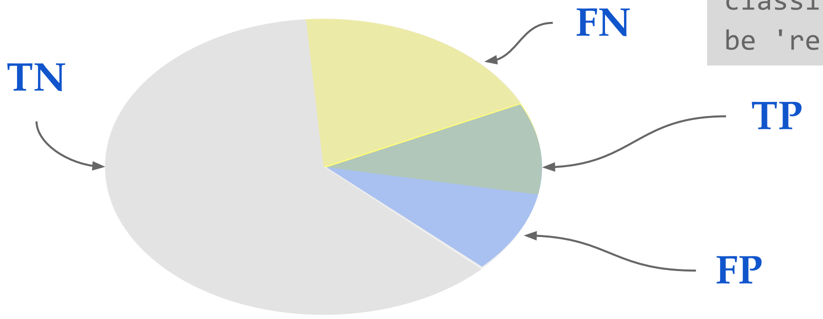
F1-measure is a combination of both

We want high precision and recall.

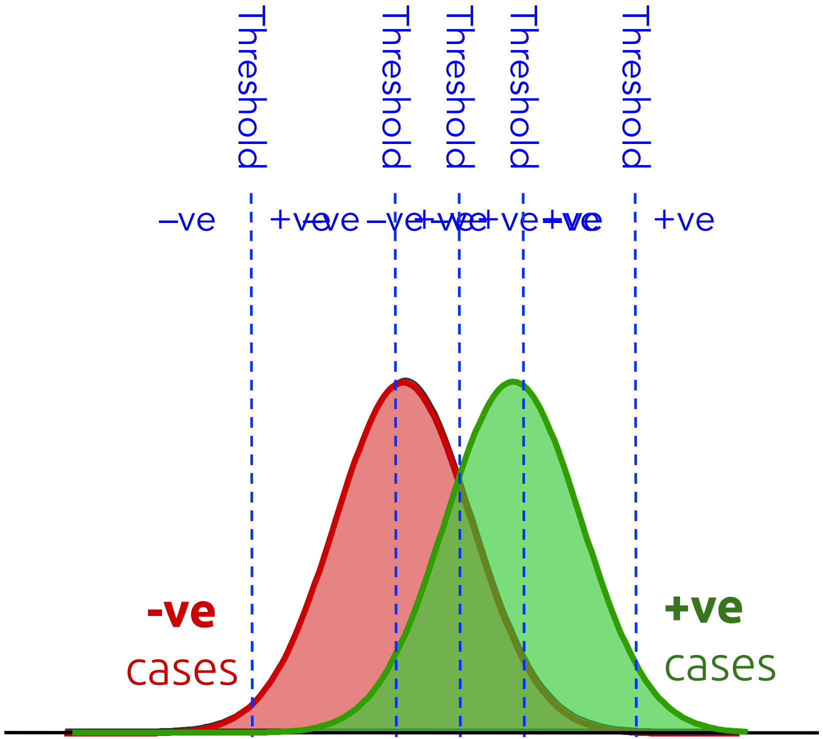
Could just use the product, but a nicer scaling is given by:

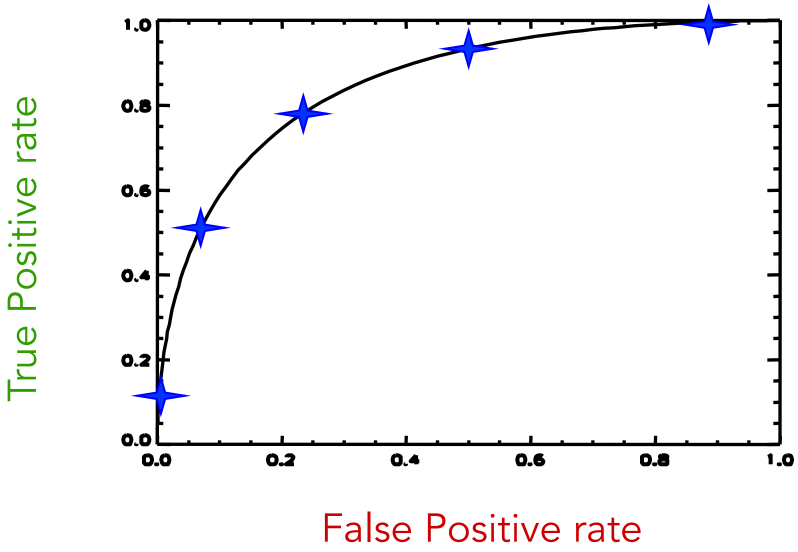
$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

		ground truth	
		P	N
predicted	P	TP	FP
	N	FN	TN



ROC curve - the idea



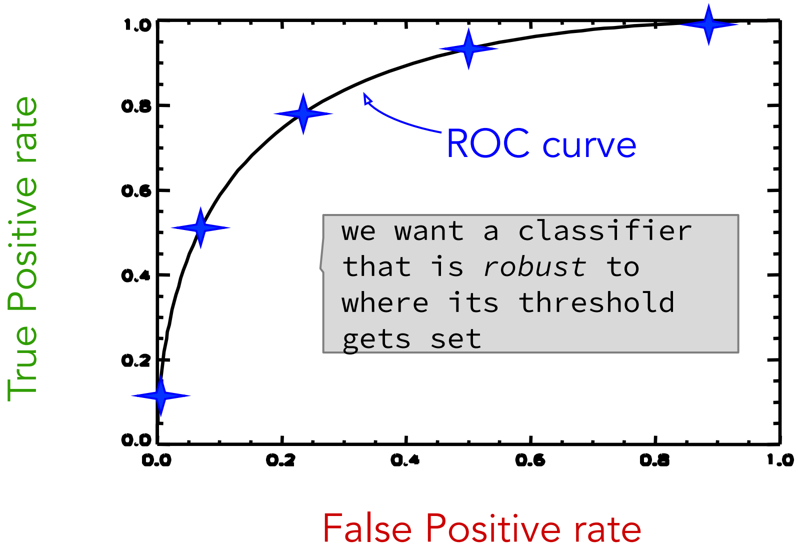


Receiver Operating Characteristic (ROC) Curve

`a graphical plot that illustrates the diagnostic ability of a binary classifier system, as its discrimination threshold is varied`

"Following the attack on Pearl Harbour in 1941, the US army wanted to increase the prediction of correctly detected Japanese aircraft from their radar signals. To do so they measured the ability of a radar receiver operator to make these important distinctions, which was hence called the Receiver Operating Characteristic"" (wikipedia)

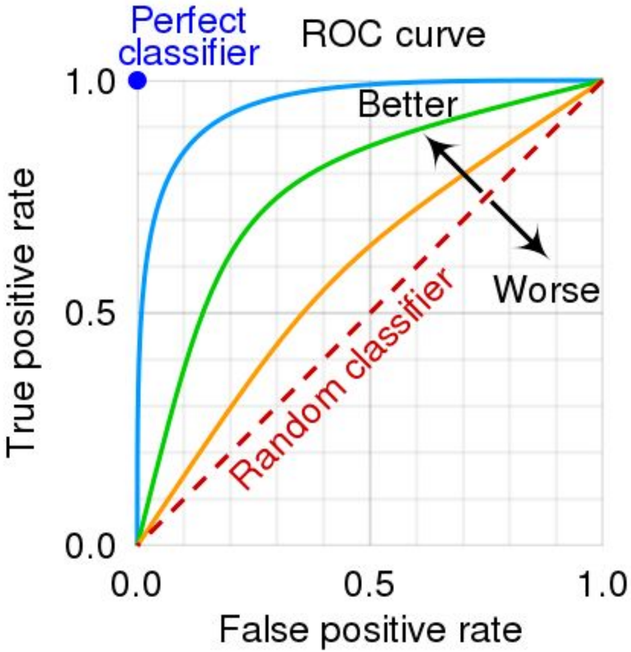
`so what's the` ***`best`*** `point on my ROC curve?`



The area *under* the ROC curve (the "AUC")

Which of these two classifiers is the best?

Compare their ROC curves



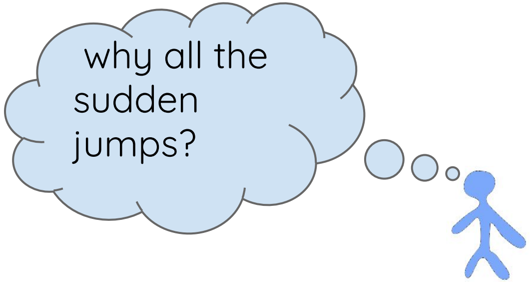
⇒ the Area under the ROC curve (AUC) gives an overall measure of a classifier's performance

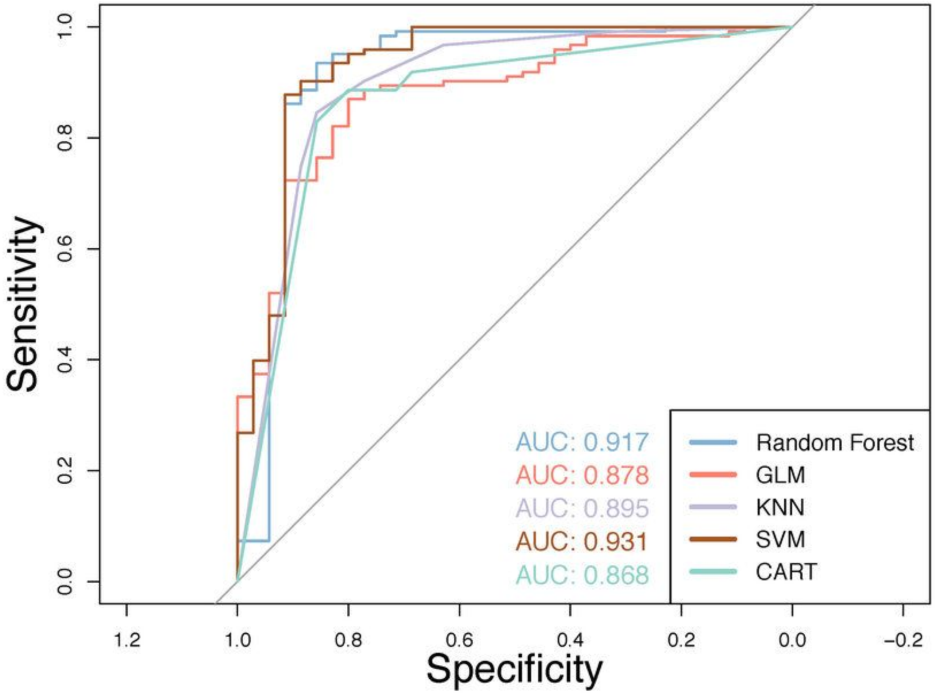
example ROC curve

(notice the weird x-axis:

The false positive rate is the same as 1 - specificity

so this is the same as ROC plot)





example: just a figure from https://www.researchgate.net/publication/331417502_Use_of_Machine_Learning_for_Diagnosis_of_Cancer_in_Ovarian_Tissues_with_a_Selected_mRNA_Panel

we can’t have a unified metric

“Why are there so many different metrics?! it’s annoying…”

1. context matters: practitioners will care about some aspects more than others
    - e.g. a company aims to maximise effectiveness of their website. But what is “effectiveness” - how can it be measured?
    - All our metrics are proxies!.. A metric that works well for research or development might not be the best choice for a real-world application.
2. costs: Some metrics are more expensive to optimise
    - e.g. Accuracy and AUC are “expensive”
3. interpretability : Some metrics are harder to explain

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2\sum_{i=1}^{n}\sum_{j=1}^{n}x_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2n\sum_{j=1}^{n}x_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2n^2\bar{x}}$$

4. consequences: (same as 1 above?)
    - e.g. Consider the effects of minimising false positives versus false negatives
5. imbalanced data
    - data can be imbalanced, where one class or outcome is much rarer than others. Metrics like accuracy can be misleading in such cases: F1-score or AUC are better

Which classifier is best?

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC
NB	71.7	.4534	.44	.16	.53	.44	.48	.7
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63
Ripp	71	.4494	.37	.14	.52	.37	.43	.6
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7
RanF	69.23	.47	.33	.15	.48	.33	.39	.63

ranking

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC
NB	3	5	1	7	3	1	1	1
C4.5	1	1	7	1	1	7	5	7
3NN	2	7	6	2	2	6	4	3
Ripp	4	3	3	4	4	3	3	6
SVM	6	8	4	5	5	4	6	7
Bagg	8	4	8	2	8	8	8	3
Boost	5	2	2	8	7	2	2	1
RanF	7	6	4	5	5	4	7	3

this is UCI Breast Cancer – a binary Classification task

How about now?

Ranking helped