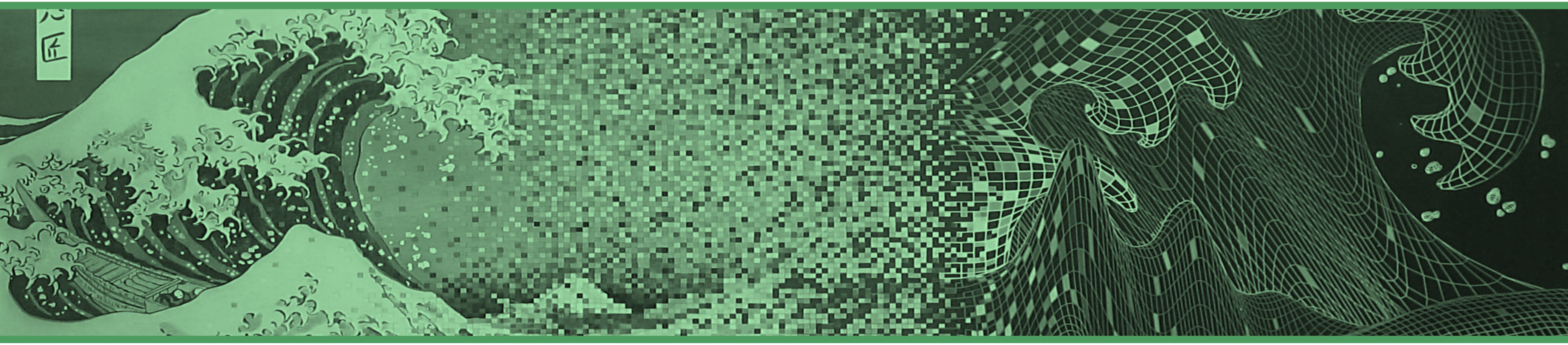
TÉCNICO LISBOA

Model Evaluation: Part I

Assessing classification models



Aprendizagem 2024

Miguel Couceiro miguel.j.couceiro@tecnico.ulisboa.pt
Andreas Wischert andreas.wichert@tecnico.ulisboa.pt

Outline

- **Introduction**
- **Evaluating classification models**
  - confusion matrices
  - accuracy, recall, F-measure
- **Cross-fold validation**
  - resampling options
  - evaluation and hyperparameterization
  - external validation

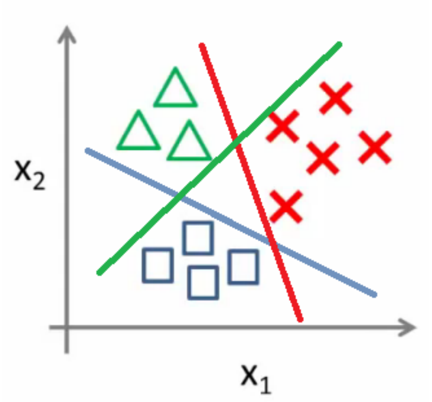
Evaluation

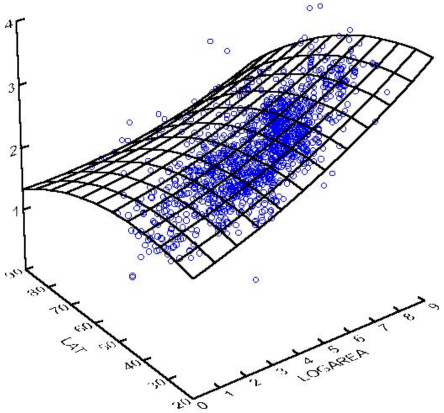
- Predictive accuracy
- Statistical significance
- Non-triviality, utility and novelty
- Robustness to noise and missing values
- Scalability (training time and testing time)
- Interpretability
- Actionability
- …

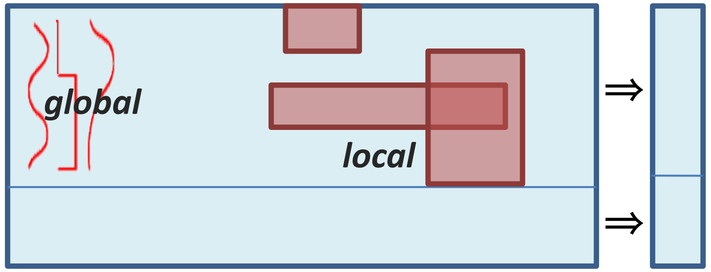
Learning functions

- Recall…
  - **predictive data modeling**
    - generally supervised
    - tasks
      - *classification*
      - *regression*

  - **descriptive data modeling**
    - can either be supervised or unsupervised
    - supervised descriptors
      - e.g. discriminant univariate rules
    - unsupervised descriptors
      - *clustering*
      - *generative models*
      - *patterns*







Evaluating learning functions

- Learning an optimal function f can be guided by loss L criteria

  $L(f^{opt}, \hat{f}_1) > L(f^{opt}, \hat{f}_2)$
  - what is a good predictor?
  - what is a good descriptor?

- Nevertheless, the learning of the models is generally separated from their posterior evaluation
  - How can I improve f? Generally restricted to available training data
  - How good is f? Generally based on unobserved observations
  - In fact, different evaluation criteria can be applied along each of these two steps

- Evaluation is essential to learn and assess
  - What is a good function of learning performance?
  - Strictly dependent on the desirable learning ends and the problem domain

Hold-out approach

- Predictors can yield good performance on training data yet poorly perform on new observations
  - problem known as **overfitting**
- We need to be able to assess learning adequacy outside training set
  - solution: set aside a separate **testing set** of observations (**hold-out**)
    - we can estimate the empirical risk on this set

$$\int_D L(z, f(\mathbf{x}))$$

- *Advantages*
  - independence from training set
  - generalization behavior can be characterized
  - estimates can be obtained for *any* classifier

Outline

- Introduction
- **Evaluating classification models**
  - **confusion matrices**
  - **accuracy, recall, F-measure**
- Cross-fold validation
  - resampling options
  - evaluation and hyperparameterization
  - external validation

Evaluating classification models

- Given a set of labeled observations, a **classification model** M is a mapping function between input variables and a categorical output variable (class variable), $M : X \rightarrow C$
  - given a new unlabeled observation $\mathbf{x}$, use M to classify: $c = M(x)$
- Learn classifier from train data and assess it against test data
  - position estimates against ground truth in a **confusion matrix**

		true/actual/target		
		P	**N**	
predicted	**P**	True Positives (TP)	False Positives (FP)	TP+FP
	N	False Negatives (FN)	True Negatives (TN)	FN+TN
		P=TP+FN	N=FP+TN	All=P+N

Evaluating classifiers: confusion matrix

- Positive class generally corresponds to:
  - case class
    - disease class (against healthy controls) in biomedicine
    - relevant documents in information retrieval
  - minority class
- Errors:
  - false positives (**type I** error) and false negatives (**type II** error)
  - one may be more interested in:
    - **minimizing FPs**, e.g. avoiding a disease diagnosis of an individual without the disease
    - **minimizing FNs**, e.g. avoiding missing a disease diagnosis if an individual has the disease
    - which of these errors are worse in the context of COVID-19 testing?

Evaluating classifiers: accuracy

- Multiple measures can be inferred from the confusion matrix
  - accuracy stance

$$\textbf{accuracy} = \frac{TP + TN}{All} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{error rate} = 1 - accuracy = \frac{FP + FN}{All}$$

  - **problems**?
    - accuracy does not disclose which error type is more frequent
      - in many domains, false positives and false negatives need to be clearly assessed
    - what if the testing observations are imbalanced towards a specific class?
      - classifiers with biases towards that same class will show misleading higher accuracy

Evaluating classifiers: beyond accuracy

- Overcoming problems of looking only to accuracy

Recall/sensitivity

– % of positive observations predicted as positive

$$recall = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Fallout/specificity

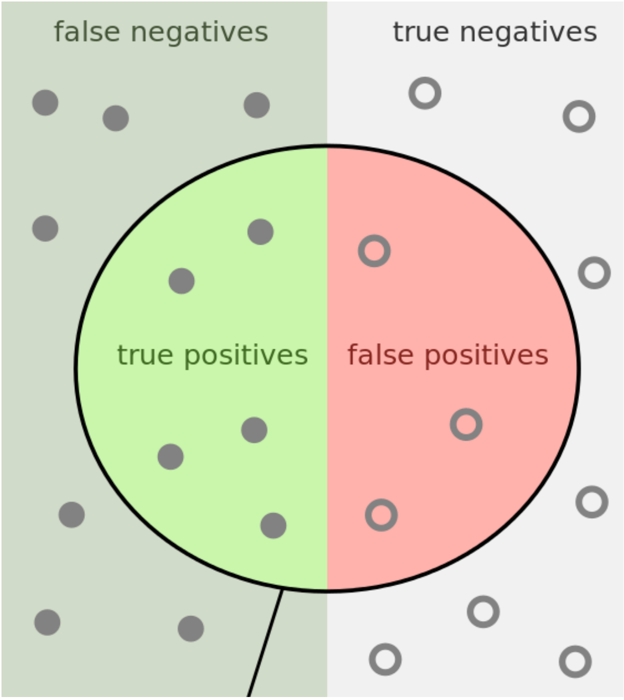
– % of negative observations predicted as negative

$$specificity = \frac{TN}{N} = \frac{TN}{TN + FP}$$

Precision

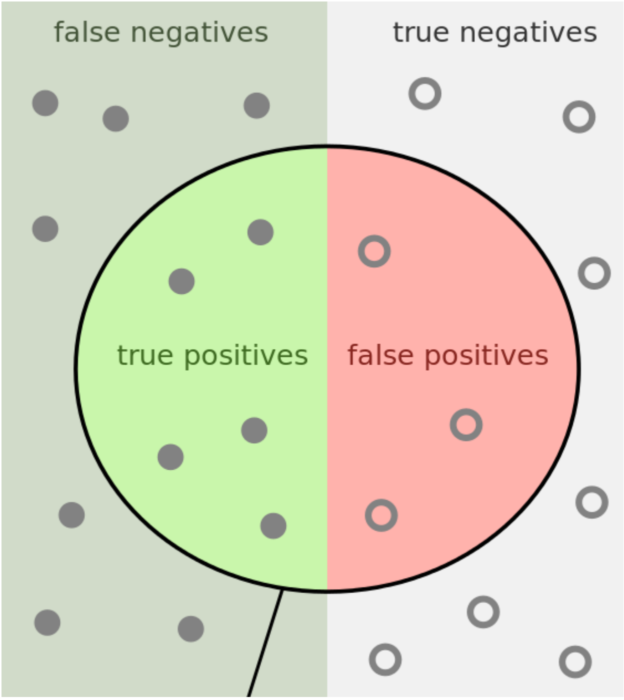
– % of positive observations among the observations predicted as positive

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



Evaluating classifiers: aliases

- Recall = TPR = Hit rate = Sensitivity
- Fallout = FPR = False Alarm rate
- Precision = Positive Predictive Value (PPV)
- Negative Predictive Value (NPV) = TN/(TN+FN)
- Likelihood Ratios:
  - LR+ = Sensitivity/(1-Specificity) ⟶ higher the better
  - LR- = (1-Sensitivity)/Specificity ⟶ lower the better
- Generally, the following measures are analysed in pairs
  - Precision / Recall
  - Sensitivity / Specificity
  - Likelihood Ratios (LR+ and LR-)
  - Positive / Negative Predictive Values



Evaluating classifiers: challenges

- We already saw that classification is generally insufficient
- Recall *versus* precision?
  - one can get high recall at the cost of low precision
    - classify all observations as positive
  - vice-versa
    - classify observations with high uncertainty as negative
- Sensitivity versus specificity?
- When working with two contrasting measures
  - 100% recall and 50% precision better than 10% and 60% precision?

Evaluating classifiers: challenges

True →	Pos	Neg
Yes	**200**	100
No	300	**400**
	P=500	N=500

True →	Pos	Neg
Yes	**400**	300
No	100	**200**
	P=500	N=500

- Both classifiers obtain 60% accuracy
- They exhibit very different behaviour:
  - on the left: weak positive recognition rate
  - on the right: weak negative recognition rate

True →	Pos	Neg
Yes	**500**	5
No	0	**0**
	P=500	N=5

True →	Pos	Neg
Yes	**450**	1
No	50	**4**
	P=500	N=5

- Classifier on the left obtains 99.01% accuracy while the classifier on the right obtains 89.9%
  - yet, left classifier labels everything as positive, missing all the negative examples

---

True →	Pos	Neg
Yes	200	100
No	300	**400**
	P=500	N=500

True →	Pos	Neg
Yes	200	100
No	300	**0**
	P=500	N=100

- Both classifiers yield same precision and recall of 66.7% and 40% (note: datasets are different)
  - yet they exhibit very different behaviour, while accuracy has no problem catching this!

Evaluating classifiers: combined measure

- Combining precision-recall (tradeoff) is **F-measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad \beta^2 = \frac{1-\alpha}{\alpha}$$

  – α ϵ [0, 1] and thus b 2 ϵ [0,∞]

- people usually use balanced **F1** measure
  – i.e. β = 1 or α = ½
  – this is the harmonic mean of P (precision) and R (recall):

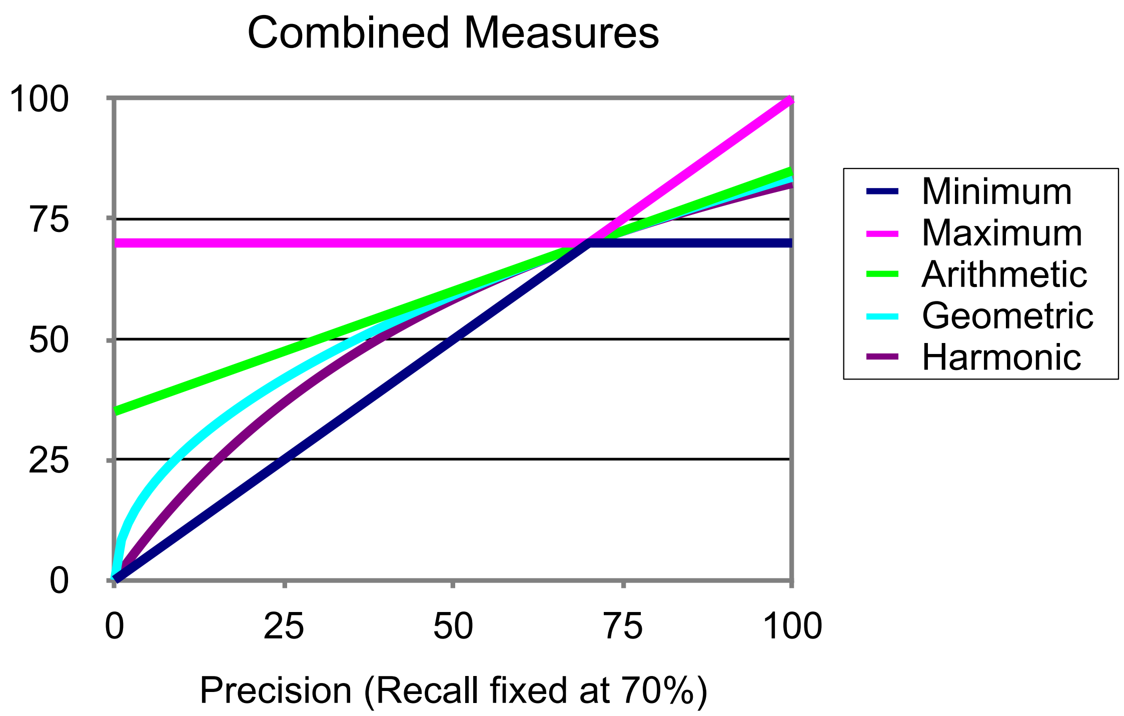
$$\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$

Evaluating classifiers: harmonic mean?

- Why don't we use a different mean of Precision and Recall as a measure?
  - e.g. arithmetic mean
- The simple (arithmetic) mean is 50% for "return everything" search engine, which is too high!

- **Goal**: punish really bad performance on either precision or recall
  - taking the minimum achieves this
  - but minimum is not smooth and hard to weight

- *F* (harmonic mean)
  - a kind of smooth minimum
  - conservative average



Evaluating classifiers: exercise

– What is F1 of the following document retrieval problem?

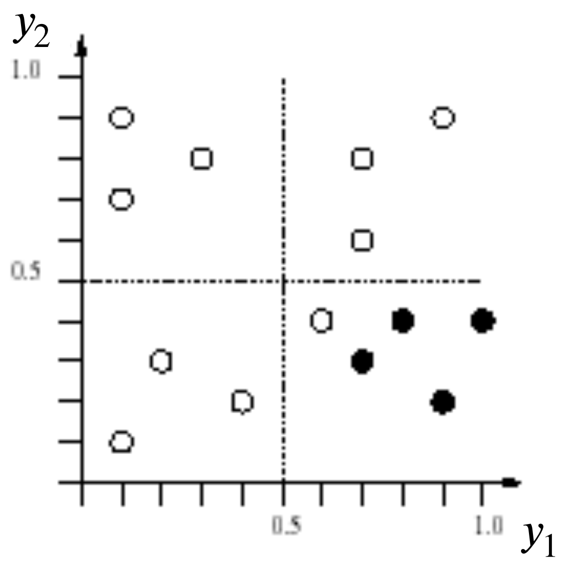
	relevant	not relevant	
retrieved	20	40	60
not retrieved	60	1,000,000	1,000,060
	80	1,000,040	1,000,120

– precision = 20/(20 + 40) = 1/3

– recall = 20/(20 + 60) = 1/4

– $$F1 = 2\frac{1}{\frac{1}{3}+\frac{1}{4}} = 2/7$$

– Compare the sensitivity and specificity of the classification boundaries



Evaluating multiclass classifiers

- Most real-world classification problems have more than two classes
  - e.g. identifying risk groups, categorizing documents, recommending products
- Extend binary confusion matrices

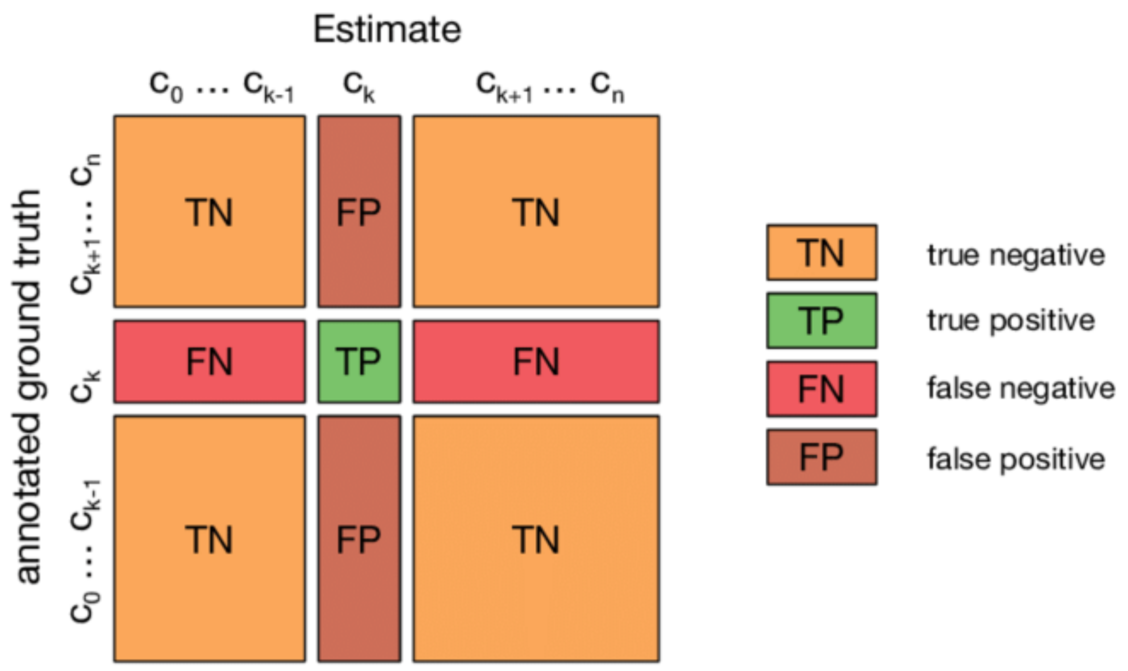
true/actual/target

		A	B	C
predicted	P	True A (TA)	False A (FA)	False A (FA)
	B	False B (FB)	True B (TB)	False B (FB)
	C	False C (FC)	False C (FC)	True C (TC)

- Accuracy is the % of observations along the diagonal

Evaluating multiclass classifiers

- Recall/sensitivity, specificity and precision *per* class
  - the target class is seen as positive
  - the negative class is the union of the remaining classes



Which classifier is better?

Real case setting

– top: absolute scores

– bottom: rankings

– contradictory views!

	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

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

acceptable contradictions

questionable contradictions

Overfitting

- One of the key properties to assess is ***generalization ability***
  - how well the model is able to correctly predict on unseen observation
  - generalization ability can be impacted by:
    - overfitting risks
    - underfitting risks

- Later on the semester we will delve with greater detail on how to assess generalization:
  - comparing training and testing accuracy
  - learning curves
  - bias and variance terms
  - external validation

Outline

- Introduction
- Evaluating classification models
  - confusion matrices
  - accuracy, recall, F-measure
- **Cross-fold validation**
  - **resampling options**
  - **evaluation and hyperparameterization**
  - **external validation**

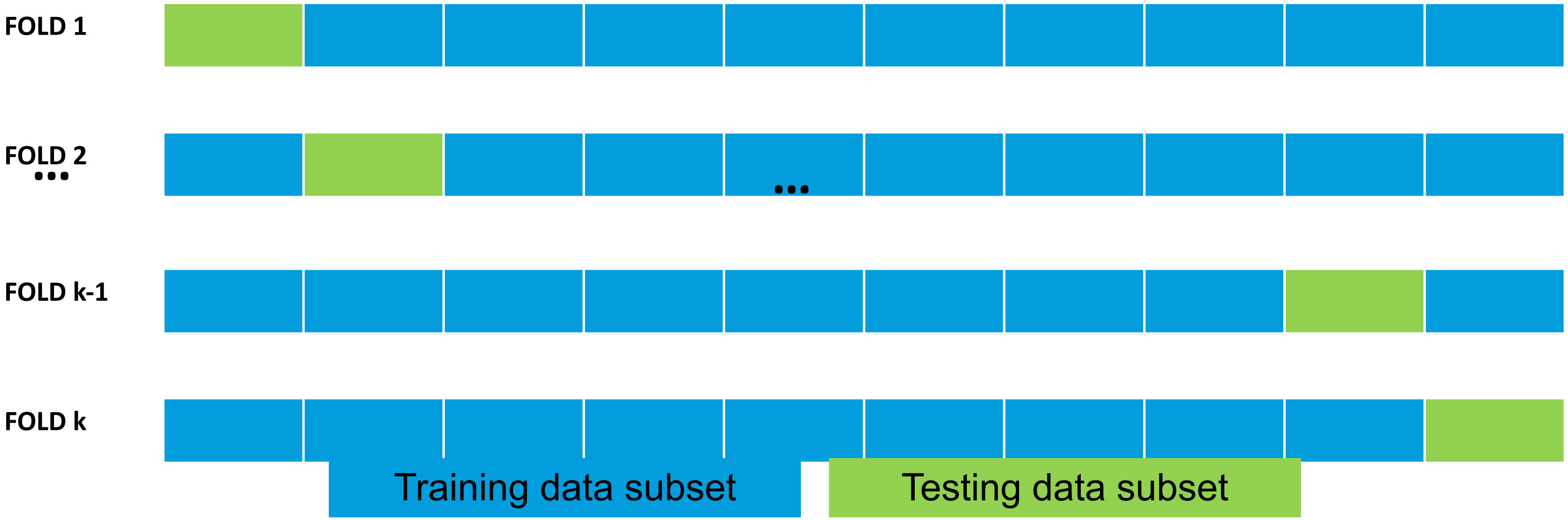
On the need for resampling

- Hold-out approach: separate training and testing data
  - problems on small-to-moderate sized data?
    - too few training examples -> learning a poor classifier
    - too few test examples -> erroneous error estimates

- Solution: **resampling** to produce multiple train-test data partitions
  - allows the algorithm to train on all/most data examples
  - allows for the collection of multiple performance estimates
    - assess performance variability
    - statistically testing
  - applicable to both classification and regression models

k-fold cross validation

- Cross-validation is the widest applied resampling method
  - the data set is divided into *k* folds
  - at each iteration, one different fold is reserved for testing while all others used for training
  - performance: average and standard deviation across folds



Cross validation: variations

- **Stratified** k-fold cross validation:
  - maintain the class distribution from the original dataset when forming the k-fold CV subsets
  - useful when the class-distribution of the data is imbalanced/skewed

- **Leave-One-Out**
  - this is the extreme case of k-fold CV where each subset is of size 1
    - also known a Jackknife estimate
  - quite effective for small-moderate dataset sizes
    - each training fold contains almost all the data
  - for large datasets, leave-one-out becomes too computer-intensive
  - also beneficial when there is wide value dispersion

Cross validation

- Example of 100 instances
  - 10-CV produces 10 folds of 10 instances each, train/test=90/10
  - 5-CV produces 20 folds of 20 instances each, train/test=80/20
- k-fold CV is arguable the best known and most commonly used resampling technique
  - with k of reasonable size, less computer intensive than Leave-One-Out
- In all the variants, testing sets are independent of one another, as required by statistical tests
- Yet, training sets are highly overlapping
  - can create a bias on error estimates (generally mitigated for large dataset size)
- Limitations
  - performance of a classifier dependent on the size of partitions (*k*)
    - while stability of estimates can give insight into robustness of algorithm, classifier comparisons in fact compare the averaged estimates (and not individual classifiers)
  - even under normality assumption, the standard deviation at best conveys the uncertainty in estimates

Multiple resampling: bootstrapping

- What can be done when data is too small for application of k-fold CV or Leave-One-Out?
- **Bootstraping**
  - assumes that the available sample is representative
  - creates a large number of new samples by drawing with replacement
    - in small data, the estimates can also have low variance owing to (artificially) increased data size
- Randomization over samples (**permutation**)
  - assess the stability of estimates over different re-orderings of the data
- **Multiple trials** of simple resampling
  - higher replicability and more stable estimates
  - multiple runs (how many?): 5x2 CV, 10x10 CV

Evaluation *versus* hyperparameterization

- Hold-out and resampling can be used for both assessment or
- Example: consider a hold-out division into training and testing data
  - **testing** set can be considered for assessment the quality of the predictive
  - training set can be further divided
    - **training set**: to learn a predictive model with specific hyperparameterization
    - **validation set**
      - to assess and choose the best hyperparameterization
      - to assess generalization ability (more in the upcoming classes)

- In fact, with the same aim we can apply nested cross-validation
  - outer cross-validation to assess model performance
  - inner cross-validation within each training fold to hyperparameterize the model

External validation

- The data can be drawn from a single or multiple **populations**
  - In clinical setting, populations correspond to:
    - individuals with different demographics
    - monitoring by different protocols, machines, hospitals
  - In utility domains, populations correspond to observations gathered from supply systems deployed on different regions or by different companies
- In classic validation
  - Hold-out and resampling is performed without care of the underlying populations
- In **external validation**
  - One or more specific populations are set aside
  - The model is learned without observations from those populations
  - Ability to generalize into unseen populations is now possible

Outline

- Introduction
- **Evaluating classification models**
  - confusion matrices
  - accuracy, recall, F-measure
- **Cross-fold validation**
  - resampling options
  - evaluation and hyperparameterization
  - external validation

Thank You

miguel.j.couceiro@tecnico.ulisboa.pt

andreas.wichert@tecnico.ulisboa.pt