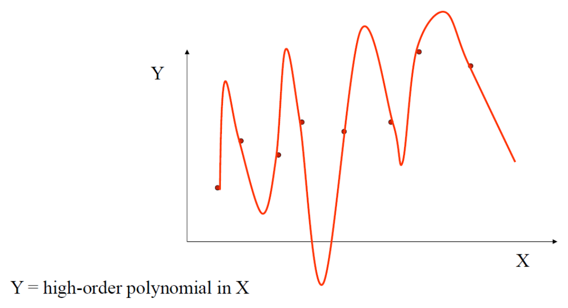
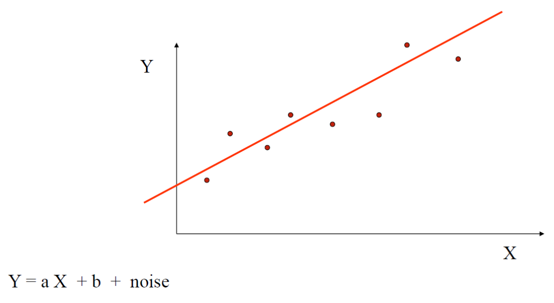
Lezione 9 13/12/2023

Evaluation issues

I dati nello spazio dei campioni potrebbero essere diversi da quelle nell'insieme di training.

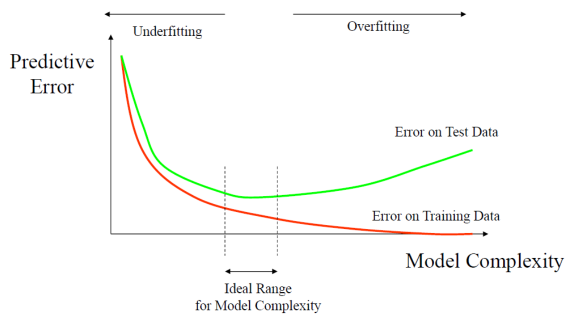
- Overfitting – fitting the training data too precisely, poor results on new data => high computational complexity!
- Underfitting – fitting the training data inaccurately, poor results on new data => low computational complexity!





Overfitting

Un modello più semplice



La linea rossa rappresenta il tradeoff tra la complessità del modello e gli errori rispetto ai dati di training.

Verde idem, ma sui dati di test.

- Natural performance measures for classification problems
  - Error rate: proportion of errors made over the whole set of instances
  - Accuracy: proportion of correctly classified instances over the whole set of instances

- True Positive, True Negative, False Positive, False Negative

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

- Machine Learning methods usually minimize FP+FN
- In practice FP and FN could have different costs
  - Medical diagnostic tests: does X have leukemia?
  - Loan decisions: approve mortgage for X?

Questa è la matrice di confusione, sulle righe abbiamo la classe reale e sulla colonna la classe predetta. Troviamo i false positive e false negatives.

Multi-class problems:

predicted→ real↓	*Class_1*	*Class_2*	*Class_3*
Class_1	94	16	10
Class_2	21	113	16
Class_3	4	4	92

- Accuracy $$\frac{TP+TN}{TP+TN+FP+FN}$$
- Precision $$\frac{TP}{TP+FP}$$
- Recall $$\frac{TP}{TP+FN}$$
- *F-measure* $$\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

Traditional (Global) Performance Measures

Given a label *l* belonging to the set of labels L:

Precision $$P(l)=\frac{\text{\# of instances correctly predicted as l}}{\text{\# of instances predicted as l}}$$

Recall $$R(l)=\frac{\text{\# of instances correctly predicted as l}}{\text{\# of instances of class l}}$$

F-measure $$F(l)=\frac{2\cdot P(l)\cdot R(l)}{P(l)+R(l)}$$

Class Level Performance Measures

In questo modo si può distinguere tra classi.

Non si usano quasi mai perché sono aggregate.

Macro-average

$$Perf^* = \frac{1}{|L|}\sum_{l=1}^{|L|} Perf(l)$$

All classes are equally important

Micro-average

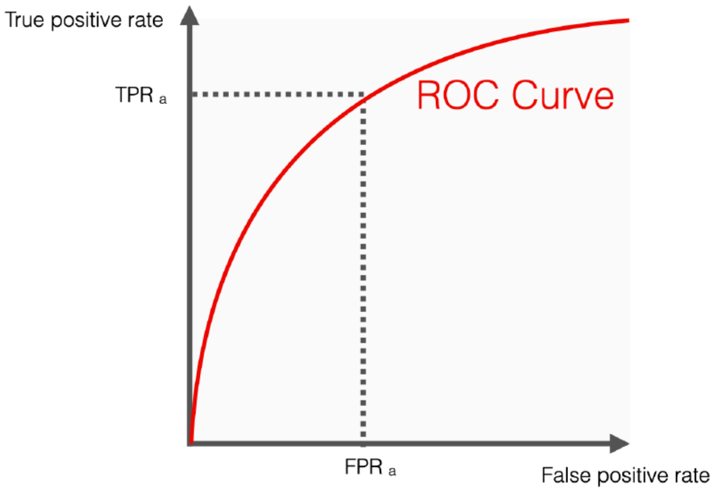
$$Perf^* = \sum_{l=1}^{|L|} \frac{|class(l)|}{\text{Tot. istanze}} Perf(l)$$

Predominant classes are more important

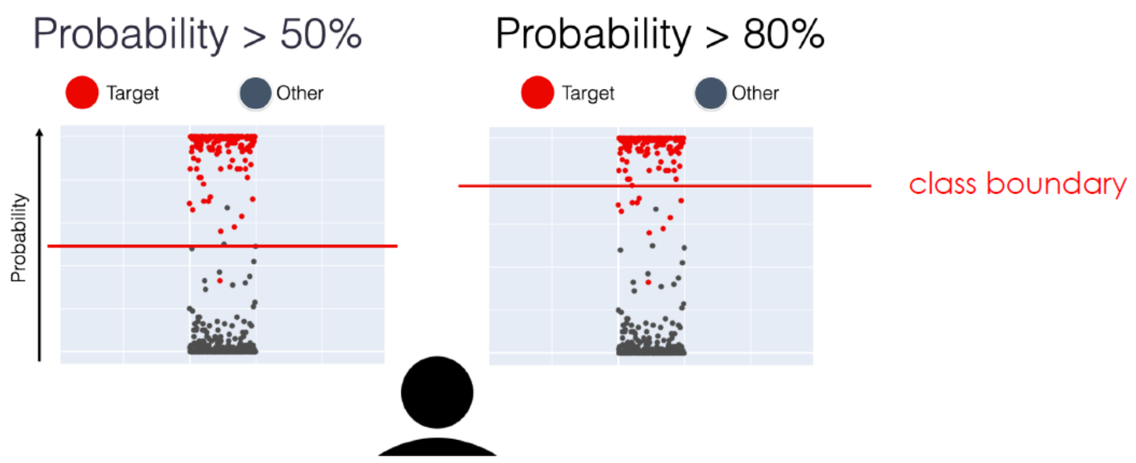
La macro-average è: ho calcolato una misura di performance, faccio la media.
La micro-average va a pesare la performance rispetto alla cardinalità degli elementi che appartengono a quella classe (quindi è pesata rispetto alla grandezza della classe).

ROC Curve

- Receiver Operating Characteristic
  - true positive rate vs false positive rate (at various **threshold** settings)

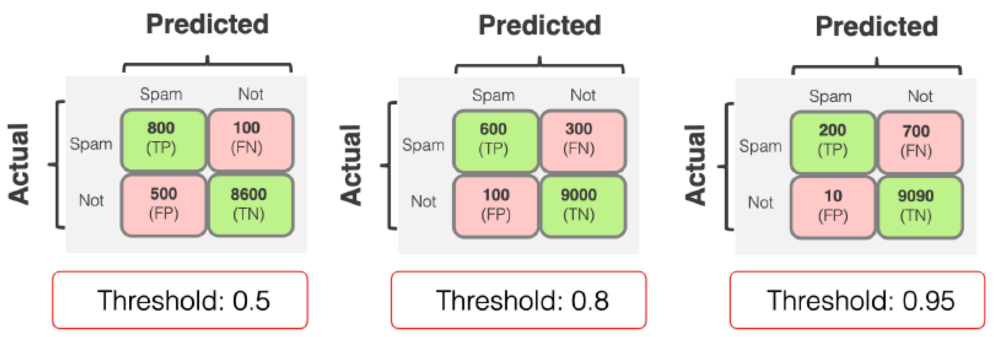


Asse x: false positive rate, asse y: true positive rate. Come cambiamo la soglia a? In un modello binario 0,5 è la soglia decisione (quando il modello deve decidere la classe di appartenenza di un dato, vede quella con probabilità più alta, ovvero sopra 0,5). Possiamo anche scegliere altri valori di threshold. Tutti i punti della curva sono la valutazione del modello con soglie decisionali che variano tra 0 e 1.



Più alzo la soglia, e più sto chiedendo la precisione, la certezza della predizione. Alzare la soglia rende il modello conservativo.

- What is the effect of changing the threshold?
- **higher threshold**: we make the model "more conservative." It assigns the True label when it is "more confident."
  - We are increasing precision, bat lowering recall.
- **Lower threshold**: we make the model "less conservative." It assigns the True label more often, even when "less confident."
  - We are increasing recall, bat lowering precision.



Threshold: 0.5

Actual \ Predicted	Spam	Not
Spam	800 (TP)	100 (FN)
Not	500 (FP)	8600 (TN)

Threshold: 0.8

Actual \ Predicted	Spam	Not
Spam	600 (TP)	300 (FN)
Not	100 (FP)	9000 (TN)

Threshold: 0.95

Actual \ Predicted	Spam	Not
Spam	200 (TP)	700 (FN)
Not	10 (FP)	9090 (TN)

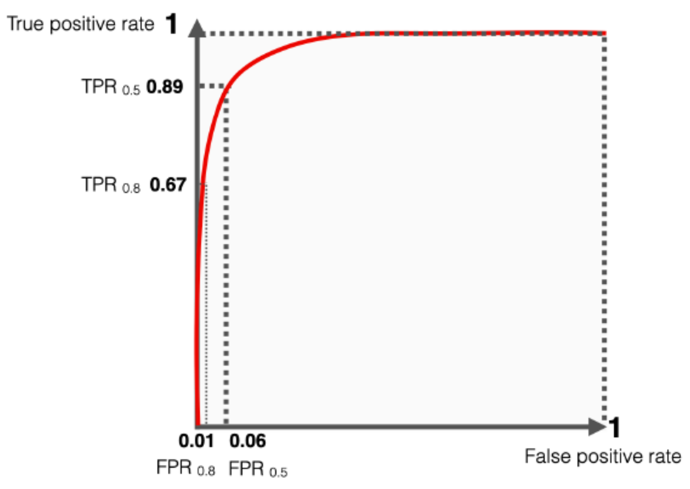
True Positive Rate

- 0.5 threshold: 800/(800+100)=0.89
- 0.8 threshold: 600/(600+300)=0.67
- 0.95 threshold: 200/(200+700)=0.22

False Positive Rate

- 0.5 threshold: 500/(500+8600)=0.06
- 0.8 threshold: 100/(100+9000)=0.01
- 0.95 threshold: 10/(10+9090)=0.001

Calcolo la curva facendo le matrici di confusione.

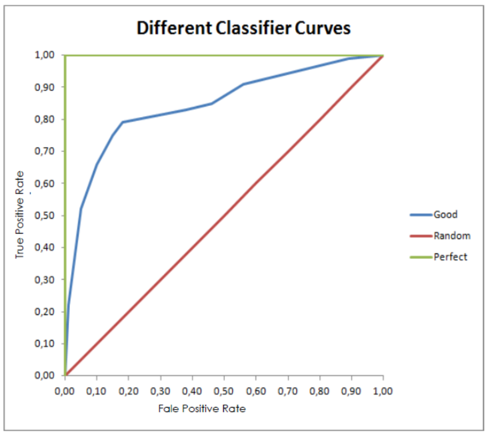


True Positive Rate

- 0.5 threshold: 800/(800+100)=0.89
- 0.8 threshold: 600/(600+300)=0.67
- 0.95 threshold: 200/(200+700)=0.22

False Positive Rate

- 0.5 threshold: 500/(500+8600)=0.06
- 0.8 threshold: 100/(100+9000)=0.01
- 0.95 threshold: 10/(10+9090)=0.001



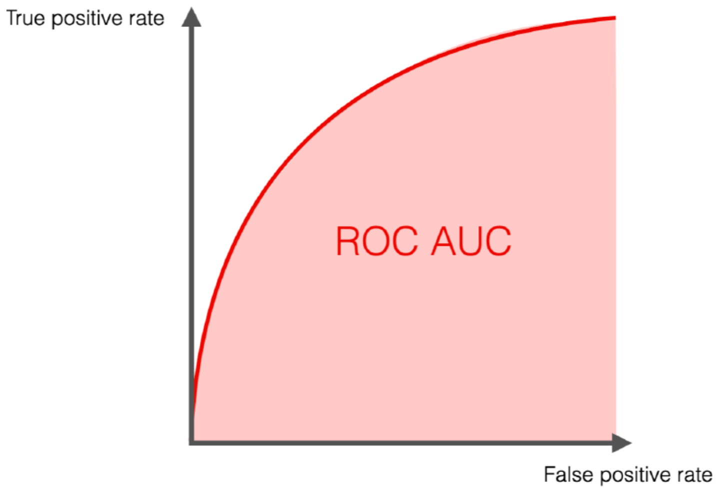
Posso usare questa curva anche per confrontare modelli:

Il classificatore randomico è sulla bisettrice. Quello perfetto invece garantisce sempre (con qualsiasi soglia) il più grande true positive rate e il più piccolo false positive rate.

AUC

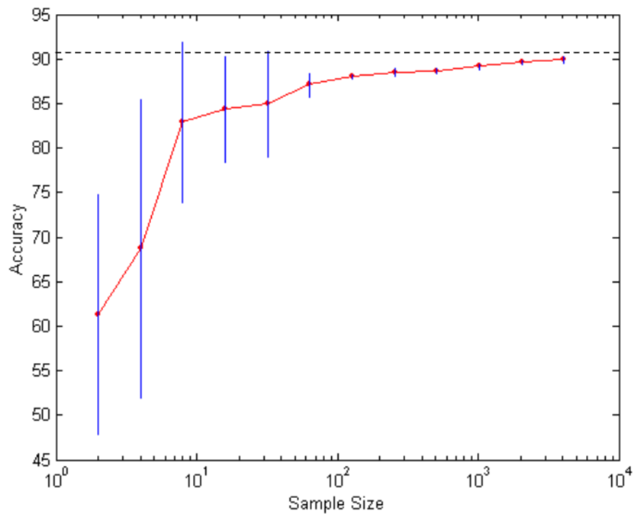
- **Area Under the Curve**

  AUC is a single scalar that summarizes the performance of a classifier across different thresholds.



- When AUC is **useful**:
  - During model training, it helps compare multiple ML models against each other.
  - Both in training and production evaluation, AUC helps to provide a more complete picture of the model performance, giving a single metric that sums up the quality across different thresholds.
- However, there are **limitations**:
  - AUC is less useful when you care about different costs of error and want to find the optimal threshold to optimize for the cost of a specific error.
  - It can be misleading when the data is heavily imbalanced (which is coincidentally often the cases where you ultimately care about different costs of errors).

Learning curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

Le curve di apprendimento mostrano il cambiamento della misura di performance (accuratezza in questo caso, si può cambiare) al variare della popolosità del campione usato durante la fase di training.

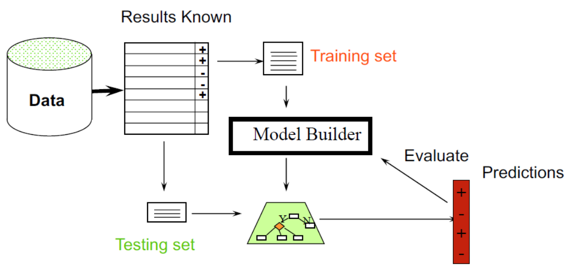
Le linee blu rappresentano la variabilità della previsione.

Evaluation on "large" data

Si considera un dataset grande quando si va almeno nelle migliaia di istanze.

- If many (thousands) of examples are available, including several hundred examples from each class, then how can we evaluate our classifier method?
- A simple evaluation is sufficient
  - Randomly **split** data into training and test sets (usually 2/3 for train, 1/3 for test)
- Build a classifier using the *train* set and evaluate it using the *test* set.

Si assume che quello che c'è in test sia molto simile a quello che c'è in test. Questo funziona sono su dataset grandi.



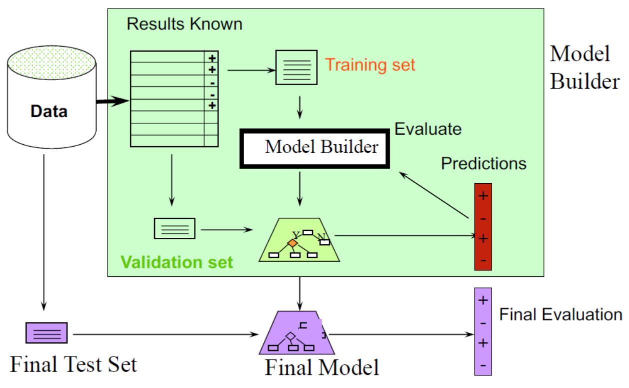
Step 1: dividi i dati tra training e test set.

Step 2: Costruisci il modello usando il training set.

Step 3: Usa il test set per valutare il modello.

- Once evaluation is complete, *all the data* can be used to build the final classifier
- Generally:
  - the larger the training data the better the classifier
  - the larger the test data the more accurate the error estimate
- **It is important that the test data is not used in any way to create the classifier!**
- **The test data can't be used for parameter tuning!**
- Proper procedure uses three sets: **training data, validation data, and test data**
  - Validation data is used to optimize parameters

Mai utilizzare i dati di test in quelli di training.



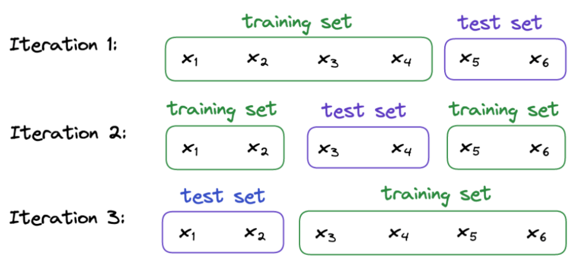
Creiamo il modello con il training set. Con i dati di validation aggiustiamo soglie, iperparametri etc. Valutiamo le performance del modello finale sul test set.

Evaluation on “small” data

- What if we have a **small data** set?
  - The chosen 2/3 for training may not be representative.
  - The chosen 1/3 for testing may not be representative.
- Solution (?): *repeated holdout*
  - repeating the process with different subsamples
    - In each iteration, a certain proportion is randomly selected for training (possibly with stratification)
    - The error rates on the different iterations are averaged to yield an overall error rate
- **We must prevent overlapping!**

In questo caso non conviene splittare il set tra training e set. Si ricorre all’holdout ripetuto, ovvero un processo che ripete questa modalità più volte.

- *Cross-validation* avoids overlapping test sets
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset in turn is used for testing and the remainder for training
- This is called *k-fold cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate



Si fanno più iterazioni in questo modo, cambiando le divisioni ed evitando overlapping sui test set.

Questo si fa per la valutazione del modello, dopo verranno usati tutti i dati per il training per creare il modello finale.

Questo processo si può ripetere cambiando il k.

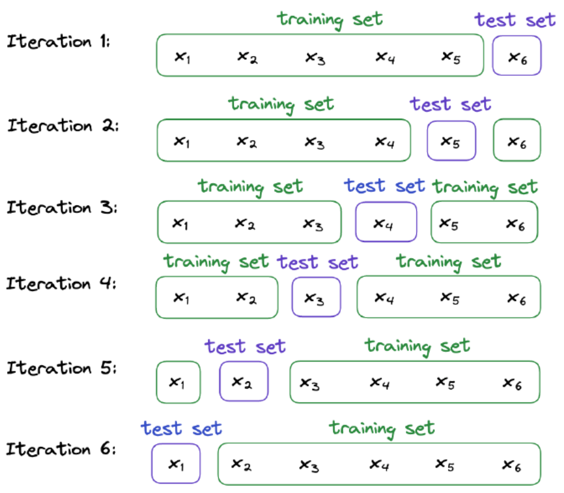
- Standard method for evaluation: *stratified* ten-fold cross-validation*
- Why ten? Extensive experiments have shown that this is the best choice to get an accurate estimate
- Stratification reduces the estimate's variance
- Even better: repeated stratified cross-validation
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

**stratified*: the distribution of selected instances, w.r.t. the class to be predicted in each fold, is similar to the original distribution of the dataset

Dataset estremamente piccoli (<100 istanze)

- *Leave-One-Out*: a particular form of cross-validation
  - Set number of folds to number of training instances
    - *n* training instances, build classifier *n* times
- Makes best use of the data
- Involves no random subsampling
- Very computationally expensive!

Usi tutti i dati tranne 1 per il training e usi l'ultimo per il test, e ripeti.



- **Confidence Intervals**
- Example: S=750 successes in N=1000 trials
  - Estimated success rate: 75%
  - How close is this to true success rate p?
    - Correct answer: with 80% confidence $p \in [73.2, 76.7]$
- Another example: S=75 and N=100
  - Estimated success rate: 75%
  - With 80% confidence $p \in [69.1, 80.1]$

Stimare l'intervallo di confidenza aiuta a capire come si posizionano, e ci dice quanto è robusto il modello.

Rappresentazione: primo caso intervallo di confidenza ampio, nel secondo caso stretto. Stesso success rate.

|————————————|

————————————|

|—|—|

Quello con l'intervallo di confidenza minore è migliore.

- Given a 10-fold cross validation, where the Entropy measures have been computed for each fold:

$$\boxed{\bar{X}(n) \pm t_{n-1,(1-\alpha/2)} \sqrt{\frac{S^2}{n}}}$$

From Experiments:
1.0422091281
0.94808406723
0.912284826627
0.951286087145
0.923315743947
1.06574821301
1.09182266828
1.01021706489
1.0638490961
1.02245326673

$$\bar{X}(n) = E(X) = E\left(\frac{\sum_{i=1}^{n} X_i}{n}\right) = \frac{10.03127}{10} = 1.003127$$

$$S^2(n) = \frac{\sum_{i=1}^{n} [X_i - \bar{X}(n)]^2}{n-1} = \frac{0.037766}{9} = 0.004196$$

The 95% confidence interval can be expressed as:

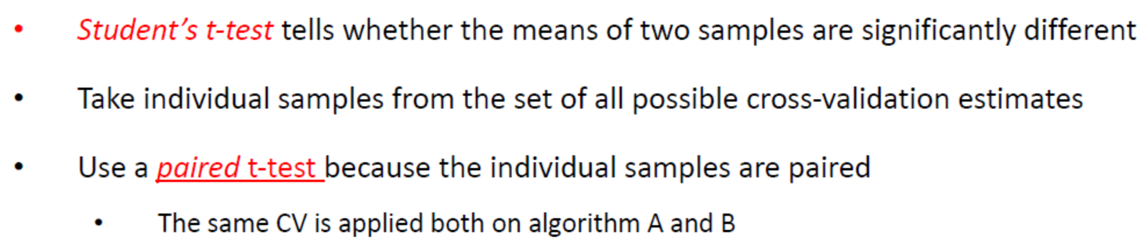
$$\bar{X}(n) \pm t_{9,0.025} \sqrt{\frac{0.004196}{10}} = 1.003127 \pm 2.262 \times 0.020485 = (0.956791, 1.049463)$$

Significance test

- *Significance tests* tell us how confident we can be that there really is a difference between to model results:

  A outperforms B?
  - *Null hypothesis*: there is no "real" difference
  - *Alternative hypothesis*: there is a difference
- A significance test measures how much evidence there is in favor of rejecting the null hypothesis
  - Let's say we are using 10 times 10-fold CV
  - Then we want to know whether the two means of the 10 CV estimates are significantly different

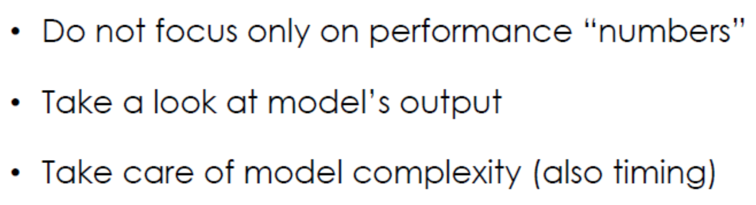
Paired t-test



- *Student's t-test* tells whether the means of two samples are significantly different
- Take individual samples from the set of all possible cross-validation estimates
- Use a *paired* t-test because the individual samples are paired
  - The same CV is applied both on algorithm A and B

Si utilizza il test paired quando (se confronto due modelli A e B) se utilizzo la stessa composizione di fold per entrambi i modelli. Altrimenti se non posso controllare questo processo, e la cross validation tra i due usa fold diversi, devo usare l'unpaired t-test.

Final remarks



- Do not focus only on performance "numbers"
- Take a look at model's output
- Take care of model complexity (also timing)

Non guardare solo la performance numerica, guarda le predizioni dei modelli.

Spesso è preferibile avere performance un pelo più bassa ma con complessità molto più bassa.