Foundations of machine learning

Contents

- Definitions

-Patterns, tasks, learning methods

- Parameters, objective function
- Training, validation, testing
- Performance metrics
- Convergence, generalization

Pattern

- Numerical

-Values associated with measurable and ordinal features

-Continuous or discrete

-Represented as numerical vectors in multidimensional space

-e.g. Counts of a detector, area of a spectrum, signal intensity in an image (most physical variables)

Pattern

- Categorical

- Values associated with qualitative features or the presence/absence of a feature
- Not convertible to ordered numerical values
- e.g. gender, nationality

Pattern

- Sequences

- sequential patterns with spatial or temporal relationships (*features*)
- fixed or variable sequence length
- e.g. sequence of image frames (video), sequence of sounds (stream audio)

Pattern

- Other structured data

- patterns organized in complex structures such as trees or graphs
- e.g. sentences of words underlying grammatical rules, genomic data networks

Task

- Classification (*classification / detection*)

Class = set of patterns with common properties

-Training a function capable of performing the mapping from the pattern space to the class space

- Assigning (*predicting*) a *pattern* to a *class*

e.g. peak signal, scattered signal, different signal detection positions and times

- 2 classes: binary classification
- more than 2 classes: multi-class classification

Task

- Regression

-Training a function capable of finding the relationship between features and the continuous value of a class

-Assigning (predicting) a continuous value to a numerical pattern

-Not to be confused with interpolation (but similar as a task)

Example energy of a power plant, dosage of a drug administration, time of disease relapse

Task

- Clustering

-Identifying groups (clusters) of patterns with similar features

-The identified clusters can be used as classes (further grouped if necessary)

The number of clusters may not be known in advance

Example Subjects with different behaviors, identification of galaxies, tumor and healthy regions in medical or biological images

Tasks in the field of vision (images/videos)

Pattern=object in an image (static or dynamic)

- Classification

Associates the class with the object

- Localization

Determines the position of the object in a bounding box

- Detection

Associates the class and determines the position of multiple objects (in bounding boxes)

- Segmentation

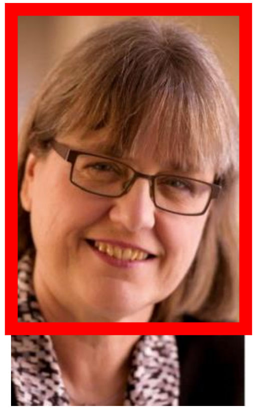
Associates individual pixels of objects with different classes (in colored pixels) and determines the position (in contours)

Tasks in the field of vision

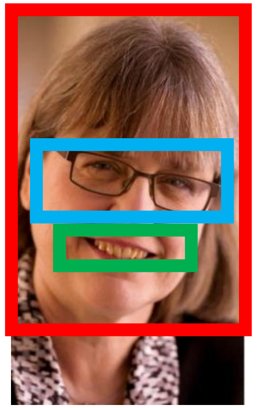
Classification

Donna Strickland (2018)

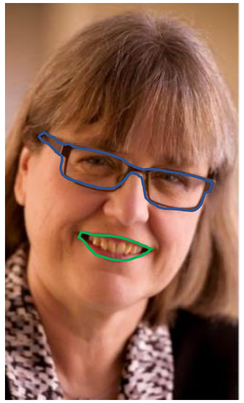
Classification + localization



Detection



Segmentation



Single object of interest (dominant)

Multiple objects of interest

Other tasks in the domain of vision (images/videos)

Pattern=object in an image (static or dynamic)

- Image quality improvement

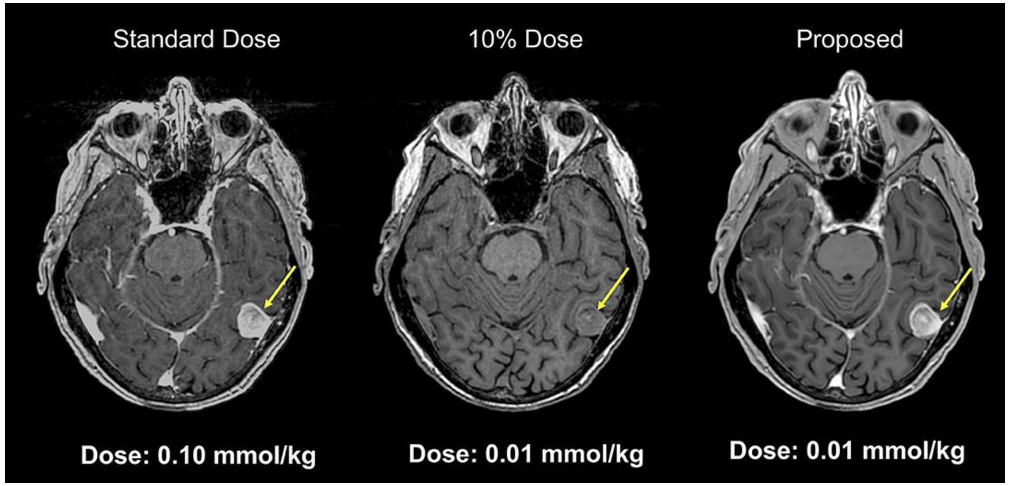
-Improves the quality of an image

- Improved image reconstruction

-Completes/speeds up the reconstruction

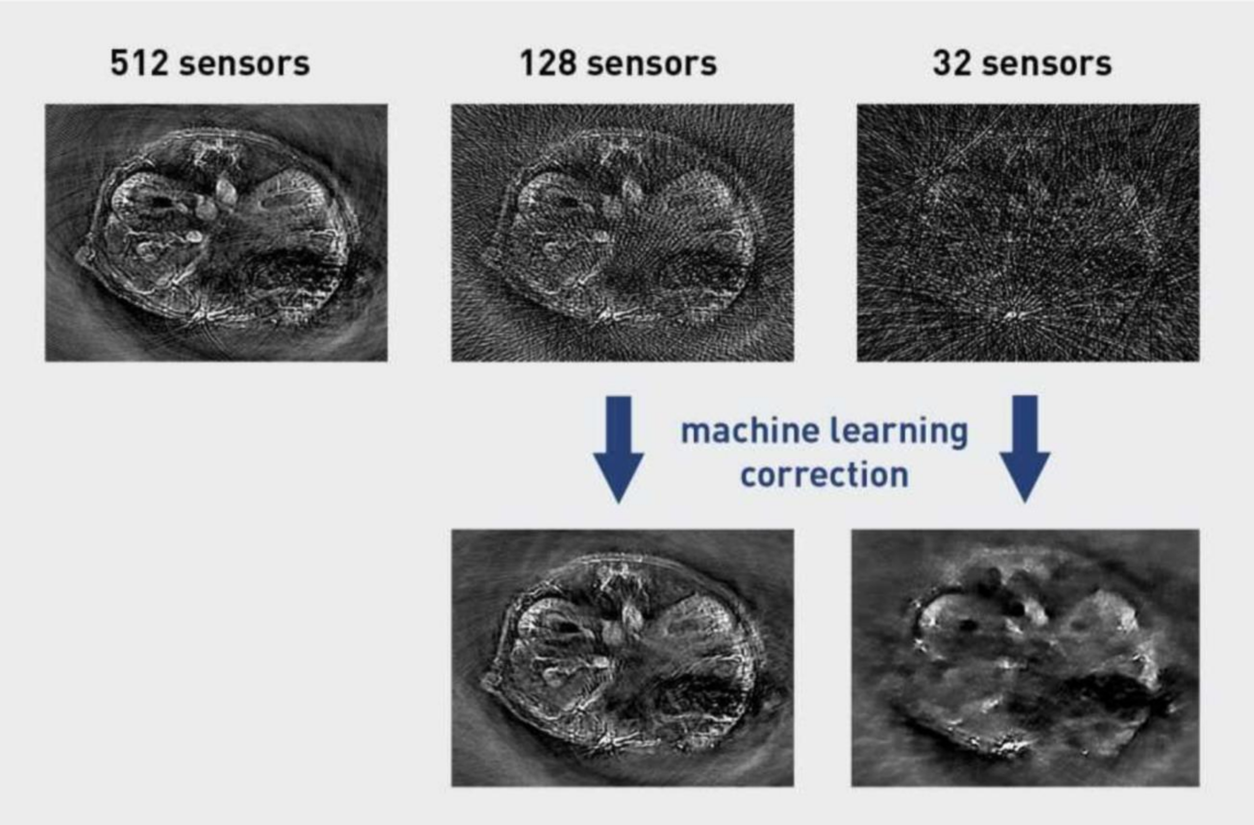
Other tasks in the domain of vision

Improved image quality



Computed tomography to improve brain contrast-enhanced signal to noise

Improved image reconstruction



Optoacoustic tomography to create cross-sectional images of a mouse

Learning methods (learning methods)

- Supervised (*supervised*)

-The classes of patterns used for learning are known a priori (typical for classification and regression tasks and in some dimensionality reduction techniques of features)

- Non supervised (*unsupervised)*

-the classes of patterns used for learning are not known a priori (typical for clustering task and in the majority of the methods for dimensionality reduction of the features)

- Semi-supervised

-the classes of patterns used for learning are partially known a priori and the distribution of associated patterns can help in optimizing the classification model

Semi-supervised

- Semi-supervised methods must make some assumptions about patterns in order to justify using a small set of classified patterns to draw conclusions about unclassified patterns. E.g:
- *Continuity assumption: It is assumed that patterns that are "close" to each other are more likely to be of the same class*
- *Cluster hypothesis: It is assumed that patterns naturally form discrete clusters, and that elements in the same cluster are more likely to be of the same class*

Algorithm Parameters

The learning behavior of a machine learning algorithm depends on a number of parameters

E.g. weights of a neural network

Learning is achieved by optimizing these parameters according to an objective function

Objective function

The objective function can be an optimization function of the algorithm (e.g. geom distance), or performance (e.g. accuracy) or error/loss/cost

It can be mathematically optimized (starting from its mathematical formula)

- *In the first case, we look for the maximum of the function (partial derivatives with respect to the parameters= 0, slope tg (gradient) >0)*
- *In the second case, we look for the minimum of the function (partial derivatives with respect to the parameters= 0, slope tg (gradient) <0)*

or with heuristic methods that modify parameters in a way that is consistent with the objective function

Heuristic methods

Methods that make it possible to predict or make plausible an outcome, which will then have to be rigorously checked and validated

A heuristic is a logically non-rigorous procedure that is accepted in view of the goal: when the urgency of arriving at a solution quickly does not allow a very careful scrutiny of each step, when one has recourse to intuition, when one is momentarily satisfied with an approximation

Training, validation , test

- *Training   Set* is the set of patterns on which the algorithm can be learned by optimizing the parameters

- *Validation Set* is the set of patterns on which to calibrate the hyperparameters

- Il Testing Set It is the set of patterns on which to evaluate the final performance.   Hyperparameters should not be calibrated on this dataset to avoid overestimation of performance

Iperparametri

Before we start learning, we need to define the hyperparameters(H)

E.g. degree of polynomial in a regression,

the number of neurons in a neural network, the error minimization parameter, the size of the images

Once these hyperparameters are fixed, the algorithm defines other parameters during learning

ES. n Random Forest Estimators, Neural Network Weights

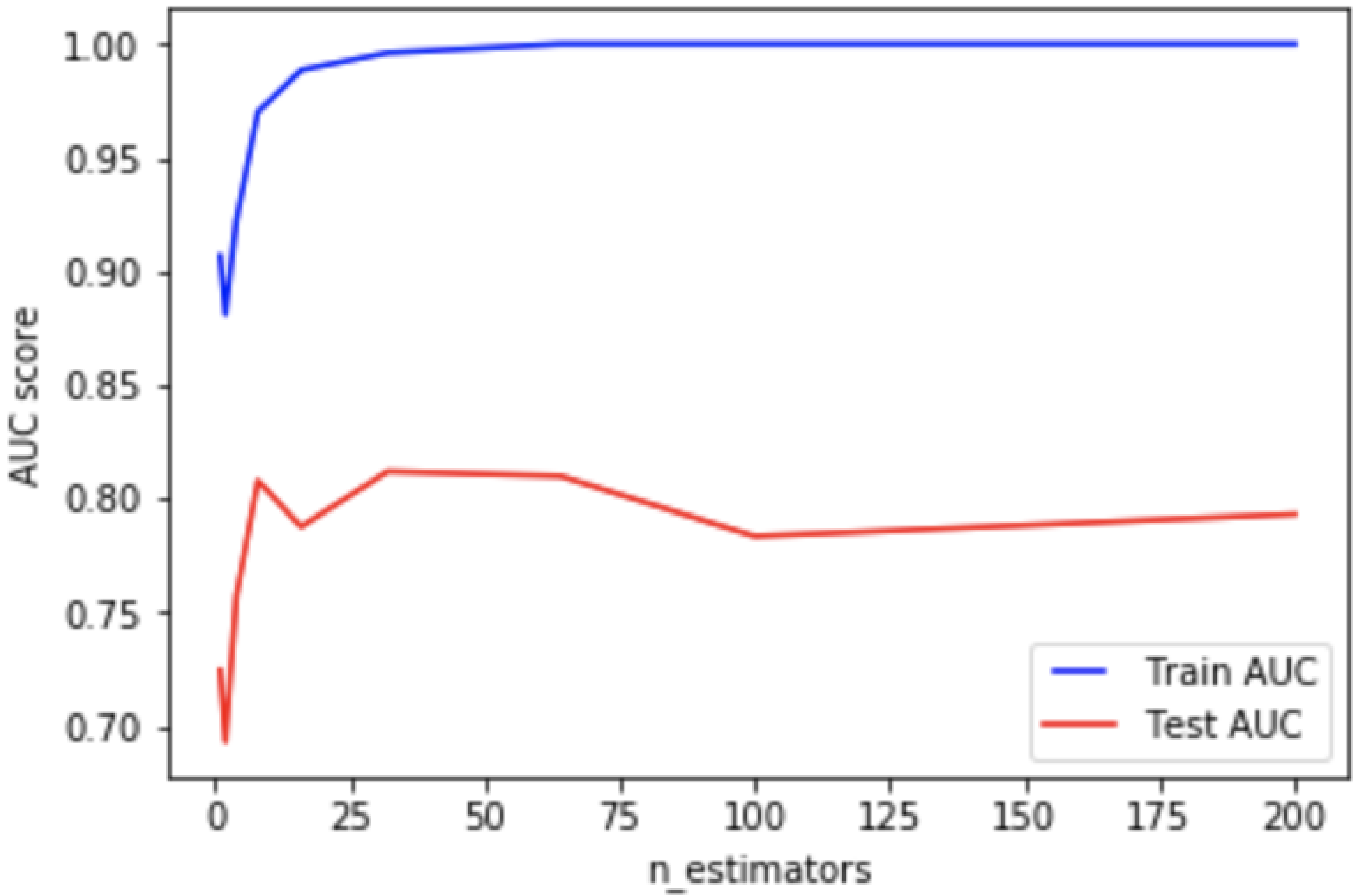
The learning time may depend on the number and type of hyperparameters defined by us before we start learning

Hyperparameter Optimization

For optimization, we can proceed with a two-tier approach: for each hyperparameter value, we perform the learning and choose the hyperparameters that provided the best performance during training (it means testing the training! Not to be confused with testing!)

The choice of values can be random (ordinally) or we can use a regression model

Parameter Optimization



Partitioning for training, validation , test

12000 pattern

- *Disjoint Sets*

- divide the patterns into 3 disjoint sets (most for training) e.g. 10000, the rest more or less equally for validation and testing e.g. 1000, 1000

- the final performance is the one on the testing sets (never seen by the algorithm)

- *K- fold cross- validation*

- k=5 partitions (fold) of 2000, you do k=5 times training and validation taking k-1=4 partitions for training each time

1 partition for validation, and the last partition for testing

-The final performance is the mean (or median) on the K=5 testing set (report the standard deviation)

Supervision

-Reference Standard

-Gold Standard

-Ground truth

Attention! It defines the performance to strive for!

Performance

- Regression Tasks

- Root Mean Square Error (RMSE) = square root of the mean squares of the deviations between true (true , target) and predicted value (predicted)

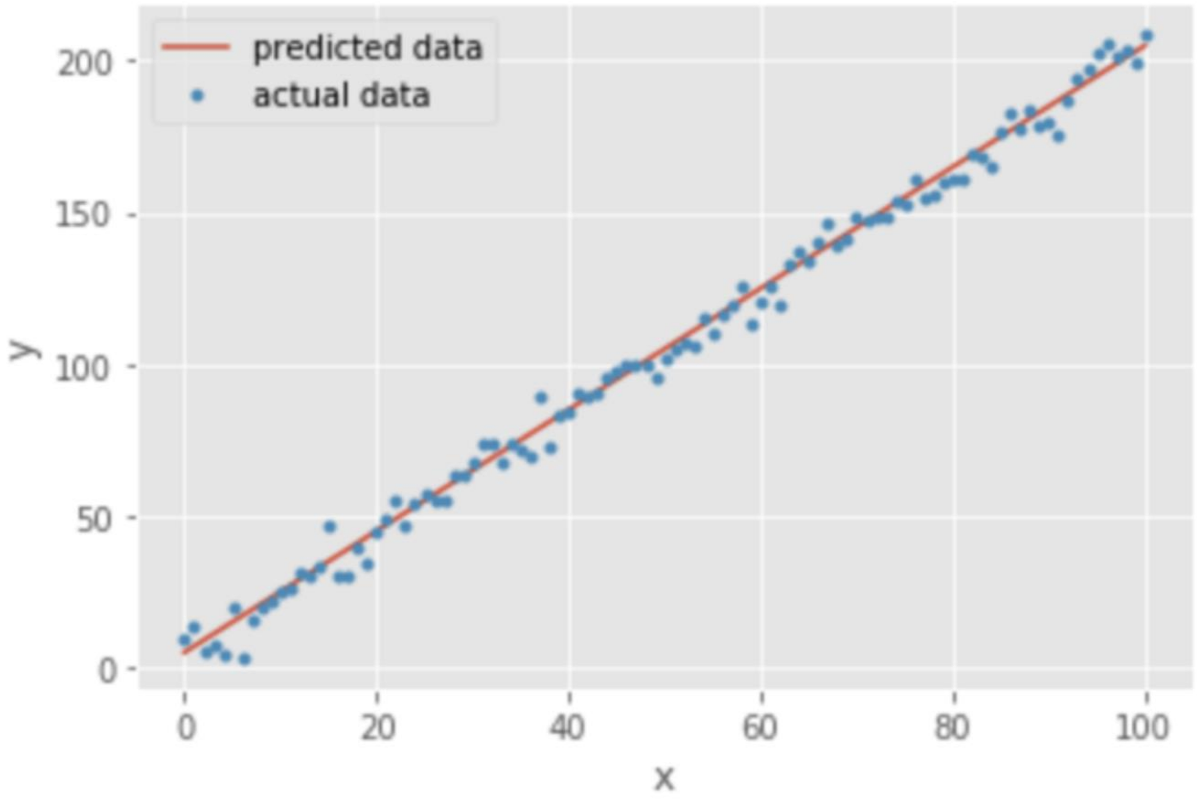
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1..N}(pred_i - true_i)^2}$$

Mean Absolute Error (MAE), Mean Squared Error (MSE)…

Performance

- Regression task

```
RMSE = 4.746304697996398
```



Performance

- Classification Tasks

- *Accuracy* = % di pattern correttamente classificati

- *Error* = 100 % - Accuratezza

- Confusion matrix

-indicates the distribution of errors in the

Each column (row) in the array represents the predicted values, while each row (column) represents the true values

Each element of the array reports the percentage of cases in which the algorithm predicted, of the class in the column, a pattern of the true class in the row

Ideally, the matrix should be diagonal.

High values (out of diagonal) indicate error concentrations.

TP e TN

Binary classifier– Class of positive patterns (P), Class of negative pattern (N)

T=P+N

- True Positive (TP): A positive pattern has been correctly assigned to positives
- True Negative (TN): A negative pattern has been correctly assigned to negatives
- False Positive (FP): A negative pattern has been incorrectly assigned to positives. Also known as Type ErrorI o False
- False Negative (FN): A positive pattern has been incorrectly assigned to negatives. Also known as d error of Tipo II o Miss

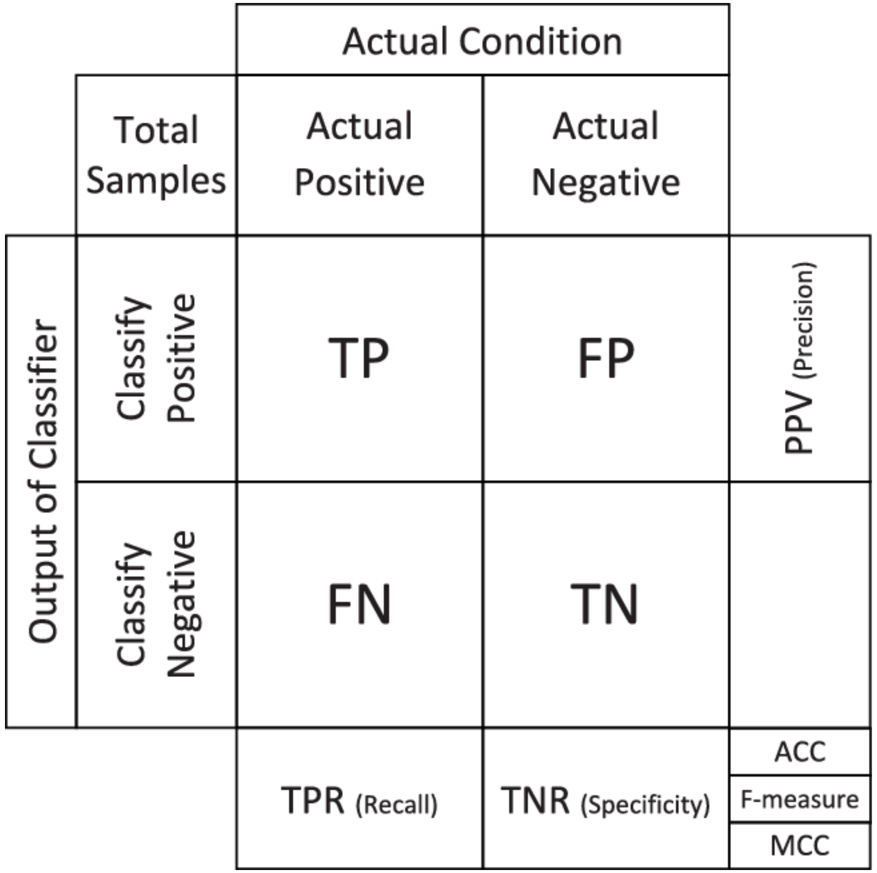
- $True\ Positive\ Rate$ (TPR) = TP/P
- $True\ Negative\ Rate$ (TNR) = TN/N
- $False\ Positive\ Rate$ (FPR) = FP/N
- $False\ Negative\ Rate$ (FNR) = FN/P
- $Accuracy$ = $(TP+TN)/(P+N)$= ($TP+TN$)/T
- $Precision = TP/\ (TP + FP)$
- Sensitivity = $Recall =\ TP\ /\ (TP+FN) =$ TP/ P = TPR
- Specificity =TN/(TN+FP)
- F_{β} score = (1+ β^2) (Precision – Recall) / [(β^2 Precision) + Recall] (media armonica di Precision e Recall per β =1) (1 is best)

Other indicators

- Positive likelihood ratio = Sensitivity (I- Specificity)
- Negative likelihood ratio = Specificity (I- Sensitivity)

Performances

Confusion matrix



<table>
<tr><td></td><td></td><td colspan="2">Actual Condition</td><td></td></tr>
<tr><td></td><td>Total Samples</td><td>Actual Positive</td><td>Actual Negative</td><td></td></tr>
<tr><td rowspan="2">Output of Classifier</td><td>Classify Positive</td><td>TP</td><td>FP</td><td>PPV (Precision)</td></tr>
<tr><td>Classify Negative</td><td>FN</td><td>TN</td><td></td></tr>
<tr><td></td><td></td><td>TPR (Recall)</td><td>TNR (Specificity)</td><td>ACC
F-measure
MCC</td></tr>
</table>

PPV, NPV, and Accuracy depend on the prevalence of examples the system learns from

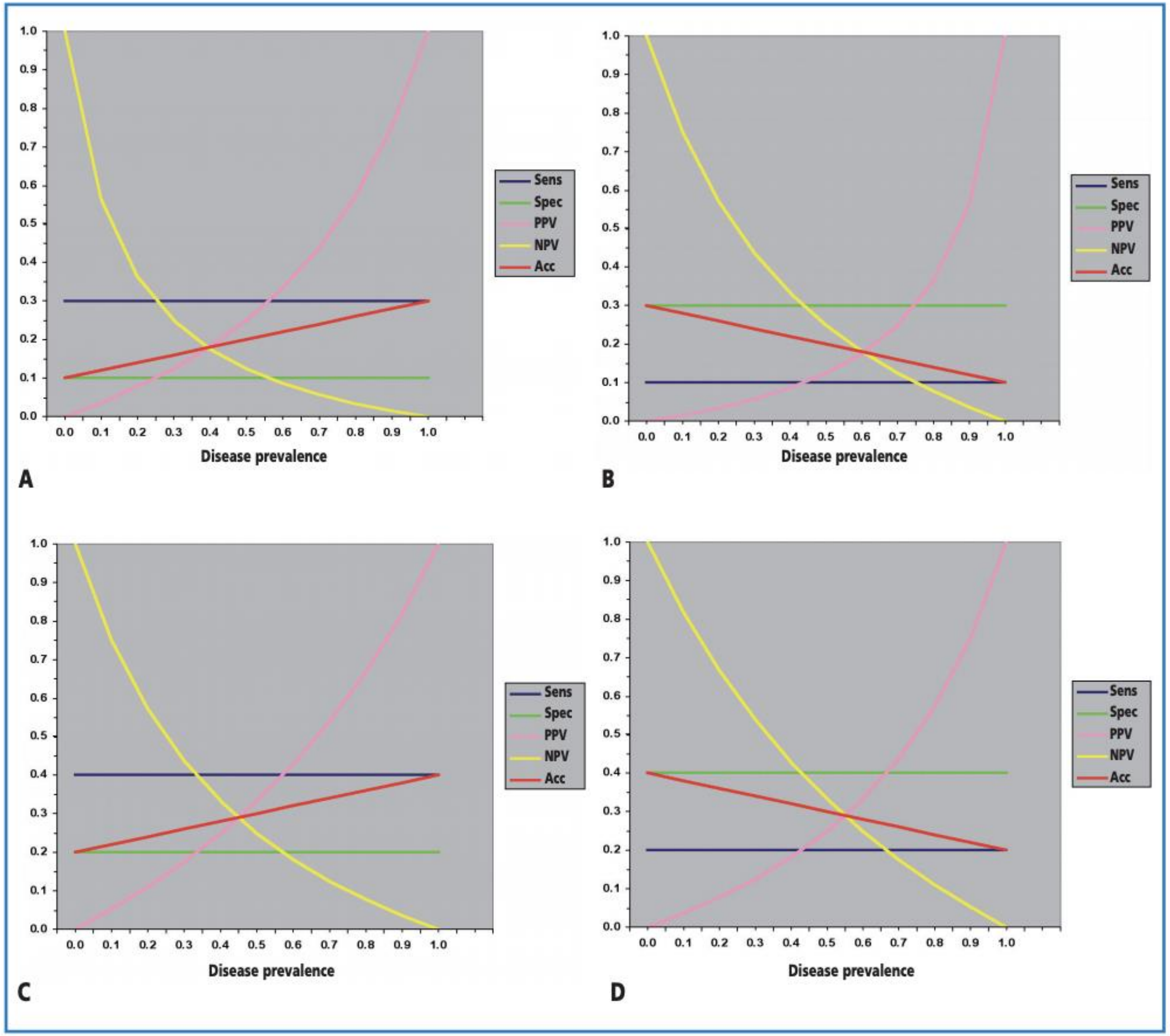
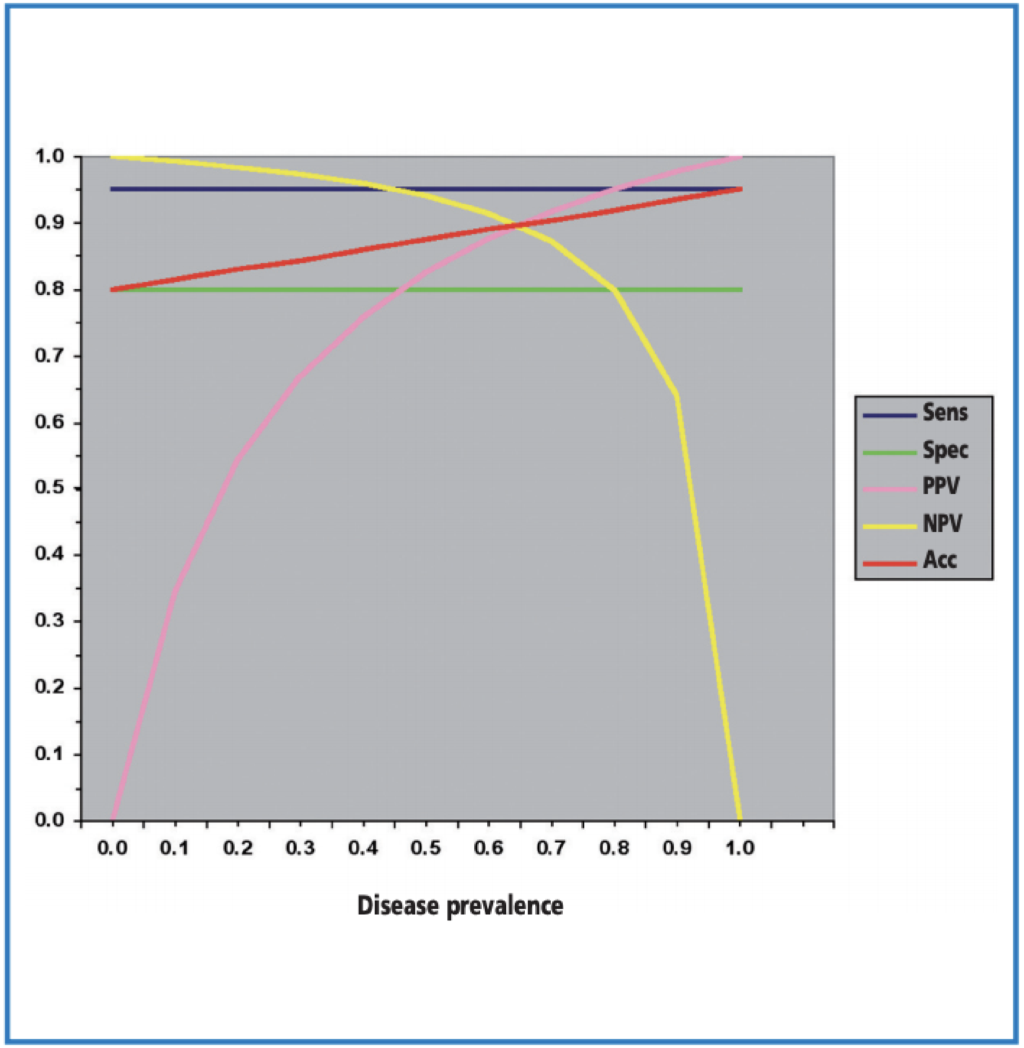


Figure 1.2. **Distribution of positive predictive value (PPV), negative predictive value (NPV), and overall accuracy as a function of disease prevalence (constant sensitivity and specificity, equal to 0.95 and 0.80, respectively).** Note that with increasing disease prevalence from 0.00 to 1.00 the predictive values change according to two different curves whereas overall accuracy increases linearly from 0.80 (specificity) to 0.95 (sensitivity). At a disease prevalence of about 0.65, overall accuracy, PPV, and NPV tend to be equal (0.89).



Binary Threshold Classifiers

Now suppose that the binary classifier makes its choice based only on the value of an index, which we call score, a function of the descriptors that characterize the object: if the score does not exceed a predefined threshold (threshold t), the object is classified in one class, if the score exceeds that threshold, the object is classified in the other class.

Es: threshold on Body Mass Index (BMI)

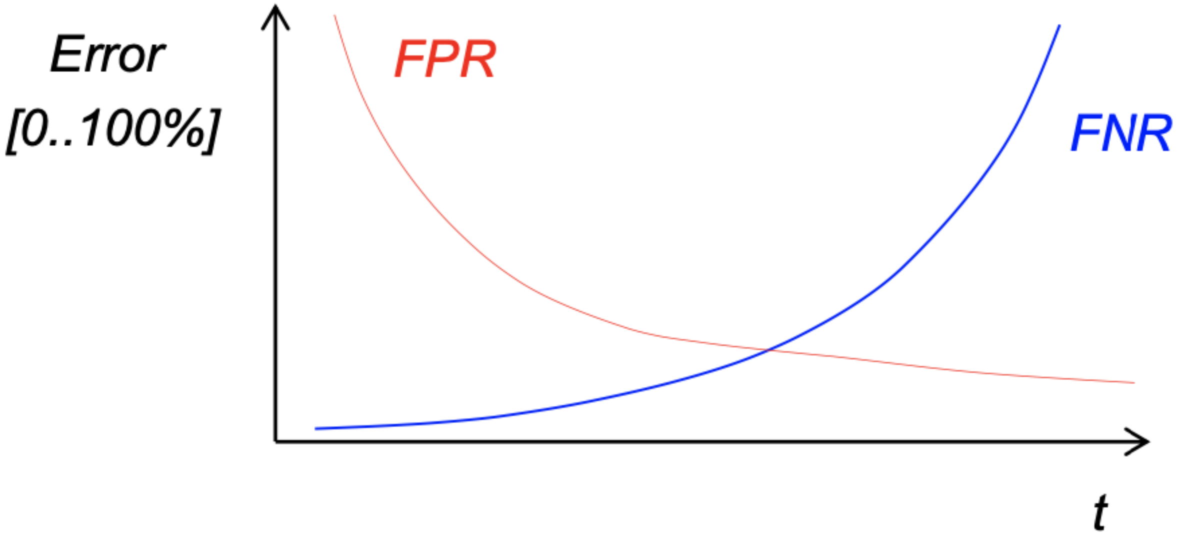
if BMI ≥ threshold then the individual is classified as obese (class 1)
otherwise the individual is classified as normal weight (class 0).

It should be noted that, depending on the value of the threshold, the same individual can be classified as sick or healthy.

Thresholds

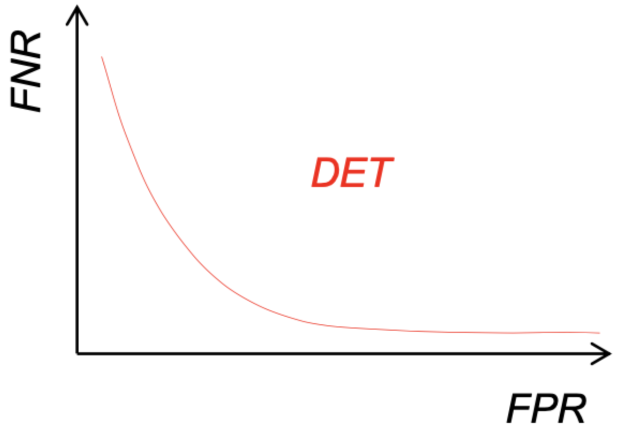
In systems with threshold (treshold, t), FPR and FNR are a function of t

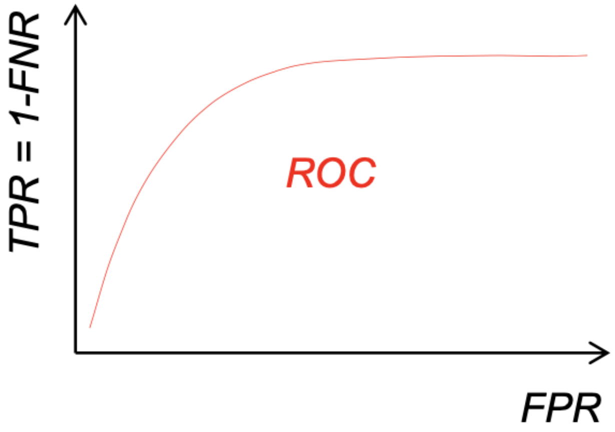
Tolerant thresholds (low t) reduce FNs at the expense of FPs, restrictive thresholds (high t) reduce FPs at the expense of FNs



DET-ROC

The two curves can be integrated into a single DET (Detection Error Tradeoff) curve or into the ROC (Receiver Operating Characteristic) which in ordinate shows TP instead of FN (inverted vertically with respect to DET)



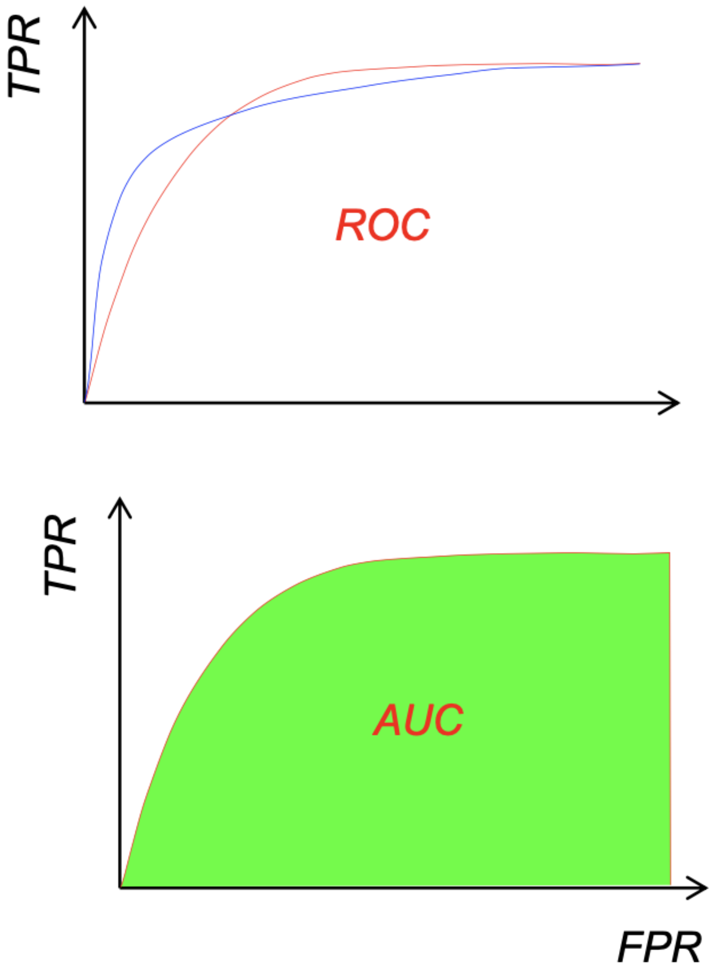


ROC-AUC

-ROC is useful because it integrates TPR and FPR but does not allow you to estimate differences in performance

- The area under the ROC curve (AUC) is a scalar (integral) in (0-1) that characterizes the average performance (the higher, the better)

- It also allows you to define a criterion to determine the optimal value of the cutoff threshold, i.e. which constitutes a good compromise between sensitivity and specificity. An example of such a criterion is to privilege sensitivity and choose as cutoff the threshold value that corresponds to the point of the ROC curve most 'at the top left' (Youden has defined an algorithm to determine this point).

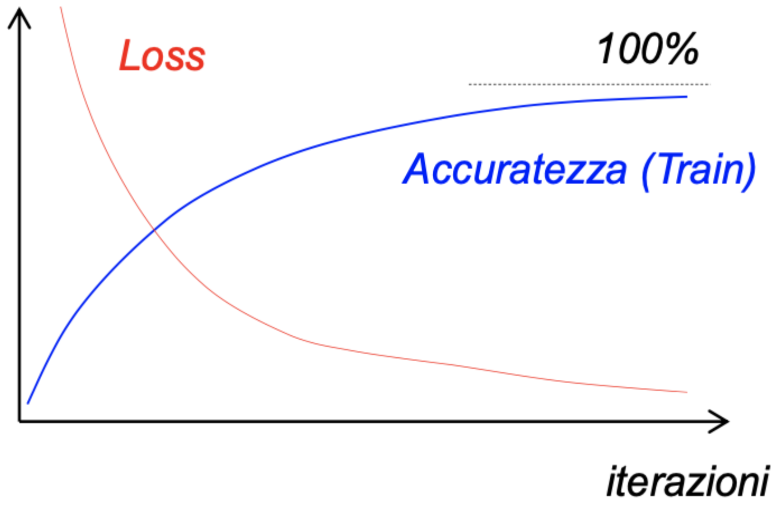


Convergence

Training is done through iterations. The first objective to be pursued during training is convergence during training.

Convergence occurs when:

- The loss (output of the target function "loss") has a decreasing trend and the accuracy has an increasing trend during training



Convergence

If the objective function does not grow/decrease or the loss function does not decrease (or fluctuates significantly) during training, the hyperparameters have been poorly defined (out of range), or the learning rate is inadequate (networks), or there are errors in the implementation of the algorithm

If the objective function increases/decreases or the loss function decreases but the accuracy does not increase, you have probably chosen the wrong objective or loss functions.

If the objective function increases/decreases or the loss function decreases, the accuracy increases but does not reach 100% during training, the classifier's degrees of freedom (express the minimum number of data sufficient to evaluate the amount of information contained in the statistic) are not sufficient to manage the complexity of the problem.

Generalization

- Ability of the model to transfer the high accuracy achieved on the Training set to the Validation set

The goal is to maximize the accuracy on the Test set

On the other hand, the goal is to maximize the accuracy of the Validation Set, assuming that it is representative of the Testing set

What should the testing set look like?

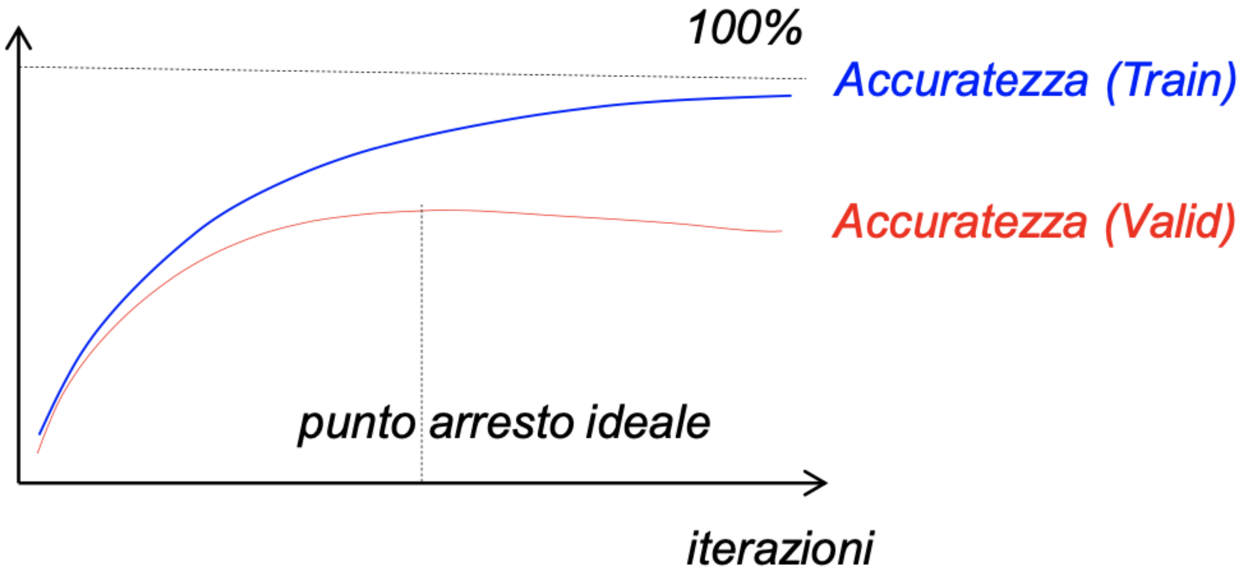
- Temporally independent
- Geographically independent….

Compromises….

Overfitting

After a certain number of iterations in which the Accuracy on the Validation Set increases, then it does not increase anymore, and may even begin to decrease due to the so-called overfitting of the model to the Training Set

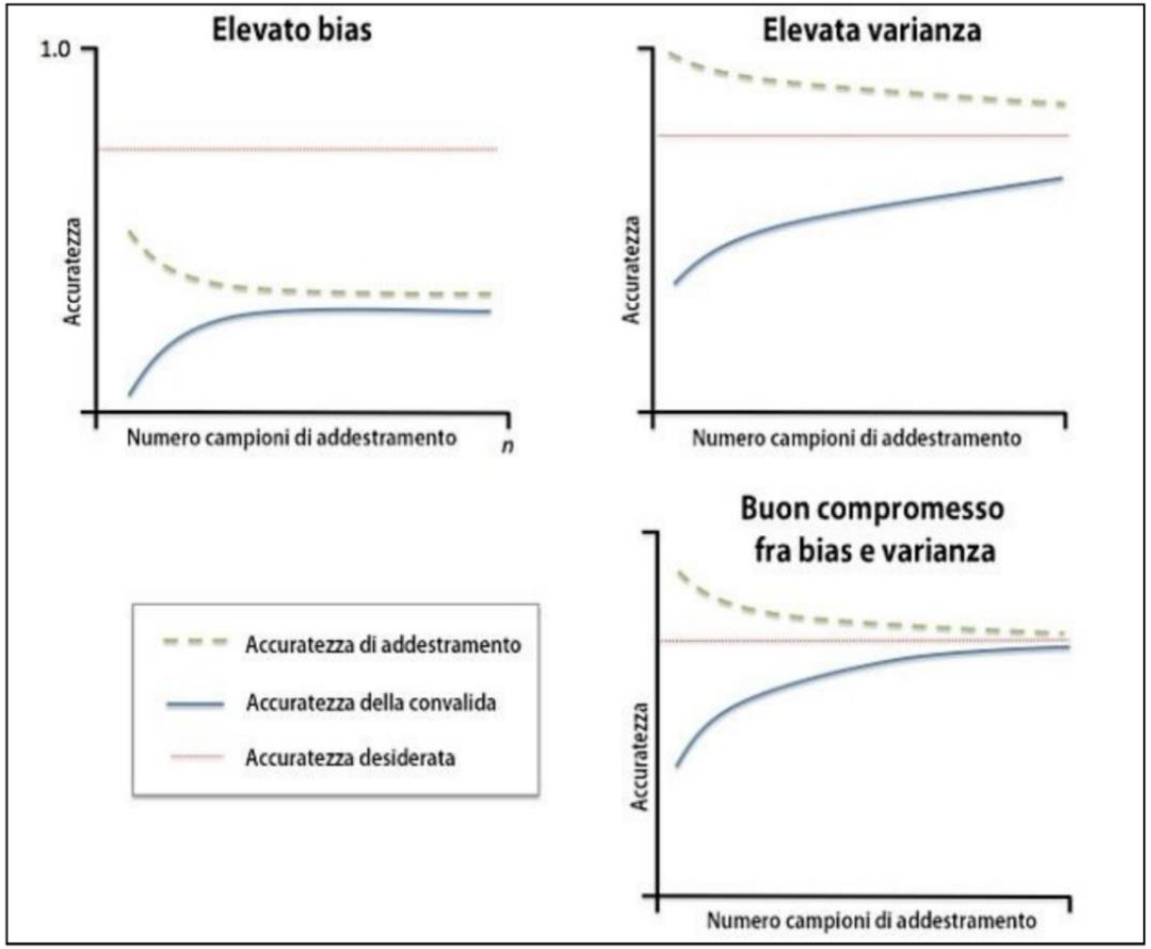
By monitoring this trend, training can be stopped



Overfitting-Underfitting

Even if Accuracy on the Validation Set increases, it may happen that..

UNDER
Few degrees
of freedom
compared to
data sets



OVER
Troppi gradi
di libertà
rispetto ai
data set

Partizioning training, validation , test

- *Nested CV*

If we use the test set to both select parameter values and evaluate the model, we risk optimistically influencing our model evaluations. Because of this, you need a different set of tests to get an unbiased evaluation of that selected model. One way to overcome this problem is to perform "nested" cross-validations. First, an internal cross-validation (equivalent to the traning/validation partition) is used to optimize the parameters and select the best model. Secondly, an external cross-validation (equivalent to the validation/test partition) is used to evaluate the model selected by the internal cross-validation.

Partitioning data into data in Training, Validation, and Test generally requires you to randomly generate the indexes of the elements to be assigned to the different sets and then proceed to subdivide them, taking care to divide the classes (in classification) or target values (in regression) in the same way