# Advanced School in Artificial Intelligence

# Introduction to Machine Learning

**Elena Bellodi**

***elena.bellodi@unife.it***

Regione Emilia-Romagna

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

FERRARIAE UNIVERSITAS · EX LABORE FRUCTUS · 13 91

Università degli Studi di Ferrara

# Outline

- Machine learning (ML) definitions
- Learning paradigms
    - **supervised**
    - **unsupervised**
    - semi-supervised
    - reinforcement
- Use of Data in ML
    - training, validation and test set
    - generalization, underfitting and overfitting
    - capacity
    - bias and variance
- Learning protocols

# Machine Learning Vocabulary *cont'd*

- **Loss function**: A function that measures the difference, or loss, between a predicted label and a true label (when available)

- **Hypothesis set:** A set of functions mapping features (feature vectors) to the set of labels

# Machine Learning Vocabulary *cont'd*

- The **hypothesis space** is defined by un underlying representation

- Each representation is appropriate for learning different kinds of target functions:
  - Linear functions
  - Logical descriptions
  - Decision trees
  - Neural Networks
  - …

# Machine Learning Vocabulary *cont'd*: Dataset decomposition

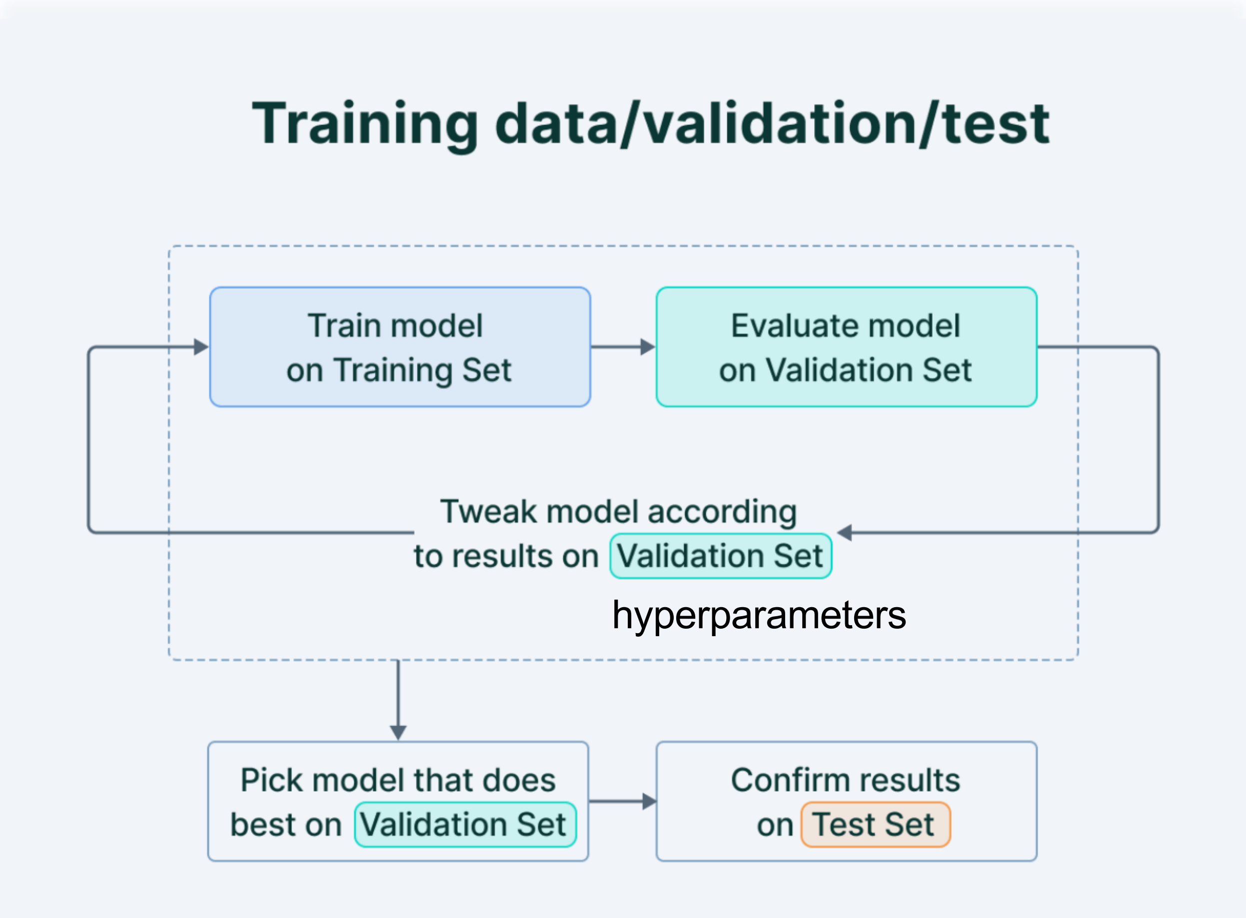
- **Training sample/set**: Examples used to **train** a learning algorithm
- **Validation sample/set**: Examples used to **tune** the parameters (*hyperparameters*) of a learning algorithm when working with labeled data
- **Test sample/set** : Examples used to **evaluate the performance** of a learning algorithm when working with labeled data

# Machine Learning Vocabulary *cont'd*: Dataset decomposition

- The **validation set** (or development set or dev set) is a set containing examples that the training algorithm does not observe
- A held-out test set, composed of examples coming from the **same distribution** as the training and testing set

- **Test examples** must not be used in any way to make choices about the model, including hyperparameters.
- Typically, one uses about 70% of the training data for training, 20% for validation and 10% for test.

# Machine Learning Vocabulary *cont'd*: Dataset decomposition



# Cross-validation

- Dividing the dataset into a fixed training set and a fixed test set can be problematic if it results **in the test set being small**
- Solution: repeat the training and testing computation on different randomly chosen subsets or splits or folds of the original dataset

## 1) *k-fold* cross-validation

- **Example: 10-fold cross – validation**
- randomly divide the data in 10 parts of equal size, and use 9 parts together for training set and 1 part for testing
- do this 10 times, using each part once for testing, and evaluate the performance every time
- at the end, compute the average test set performance

# Cross-validation

- *We are evaluating a learning algorithm k times, we are not evaluating k models* (which are the output of the algorithm)
- Once we are satisfied with the performance of our learning algorithm, we can run it over the entire data set to obtain a single model
- Cross-validation is conventionally applied with $k$ = 10, although this is somewhat arbitrary

**2) *leave-one-out* cross-validation**: set $k$ = $n$ and train on all but one test instance, repeated $n$ times. This means that in each single-instance 'fold' our accuracy estimate is 0 or 1

# Cross-validation

*3) stratified* **cross-validation**: if the class distribution is **unbalanced**, the folds are «stratified» to achieve roughly the same class distribution in each fold

By averaging over training sets we get a sense of the **variance** of the learning algorithm (i.e., **its dependence on variations in the training data**), although it should be noted that the training sets in cross-validation have considerable overlap and are clearly not independent.

# (Supervised) Learning Stages

1. Start with a given collection of examples (**dataset**)
2. Randomly partition the data into a training sample, a validation sample, and a test sample
3. Associate **relevant features** to the examples
   - Chosen by the user
   - Useful features can effectively guide the learning algorithm, while poor or uninformative ones can be misleading
4. Choose the **learning algorithm**

# (Supervised) Learning Stages

5. Training phase: use the features selected to train the learning algorithm by fixing different values of its free parameters.
6. Validation phase: For each value of these parameters, the algorithm selects a different hypothesis out of the hypothesis set. **We choose among them the hypothesis resulting in the best performance on the validation sample**
   - *This is the output model*
7. Test phase: using that hypothesis, we predict the examples in the test sample (**we predict the label** of test examples)
8. The performance of the algorithm is evaluated by using the *loss function* associated to the task, computing the error on the test sample

- 5-6-7-8 can be substituted by cross-validation

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

# (Supervised) Learning Example

- Example: **Credit approval by a bank**

| | |
|---|---|
| • Salary, debt, years in residence, ... | *input* $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$. |
| • Approve credit or not | *output* $y \in \{-1, +1\} = \mathcal{Y}$. |
| • True relationship between $\mathbf{x}$ and $y$ | *target function* $f : \mathcal{X} \mapsto \mathcal{Y}$. (The target $f$ is *unknown*.) |
| • Data on customers | *data set* $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$. ($y_n = f(\mathbf{x}_n)$.) |

- *data D are the training examples*

# (Supervised) Learning Example

- Start with a set of candidate hypotheses $\mathcal{H}$ which you think are likely to represent *f*.
  - $\mathcal{H}$ = *{h1, h2, . . . }* is called the hypothesis set
- Select a hypothesis *g* from $\mathcal{H}$. The way we do this is called a **learning algorithm**.
- Use *g* for new customers (prediction): we hope g ≈ f.
  - Ideally $g(x_n) = y_n$
  - **g** is the **output model**

# (Supervised) Learning Example

- ***X*, *Y* (sometimes) and D are given** by the learning problem
- $f$ is fixed but unknown
- We **choose** $\mathcal{H}$ and the learning algorithm
- **Future decisions** will be based on *g*, and will be good only to the extent that *g* faithfully replicates $f$