

Machine Learning & Signals Learning

7 Learning Systems

7.1 Basic Workflow

The basic ML/DL workflow is presented in Fig. 7.1. The workflow parts are:

- Goal definition
- Data: available data
- Pre-processing: preliminary dataset exploration and validation of dataset integrity (e.g., same physical units for all values of the same feature).
- Model: basic assumptions about the hidden pattern within the data
- Model training: minimization of the loss functions to derive the most appropriate parameters.
- Hyper-parameter optimization: tuning the model sub-type.
- Performance assessment according predefined metrics.

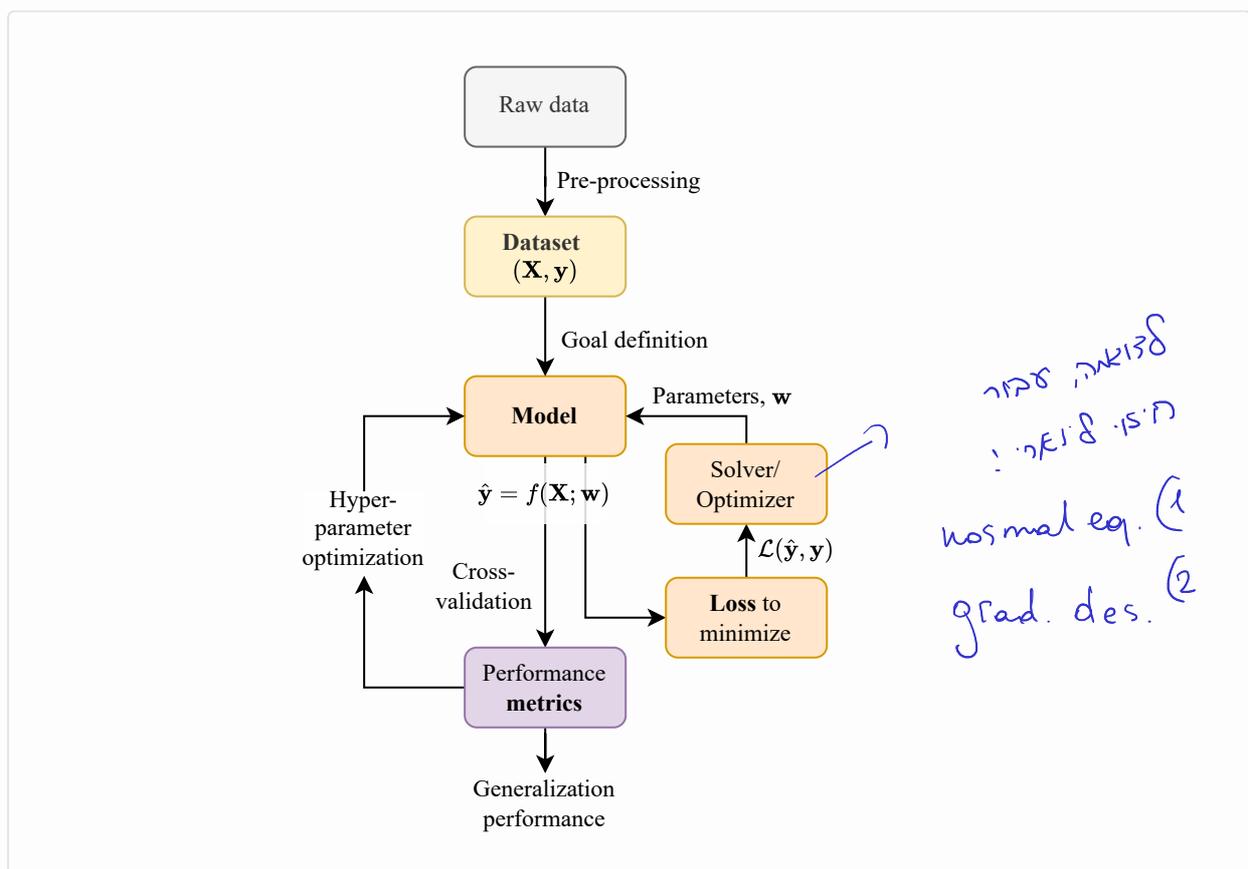


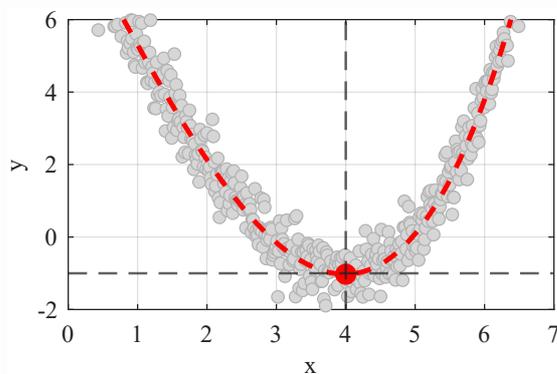
Figure 7.1: Basic workflow of ML/DL solution.

Baseline The basic end-to-end workflow implementation is called baseline.

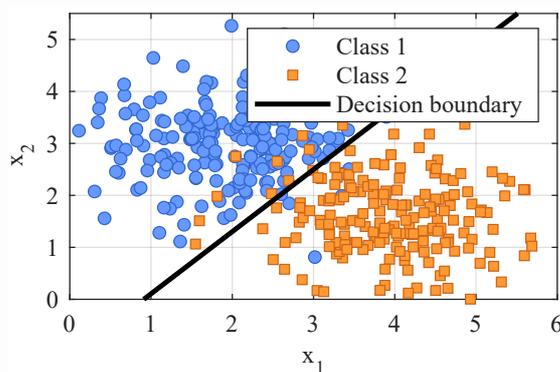
7.1.1 Goal definition

Typical related goals:

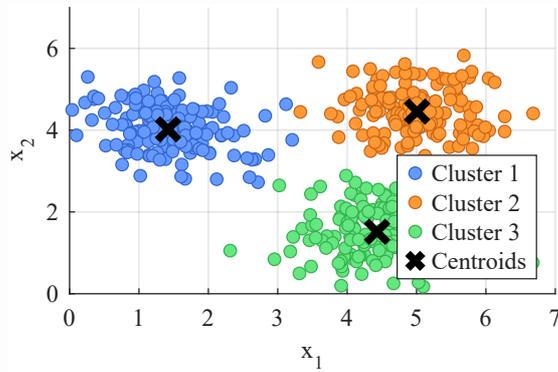
- Prediction or regression, y is quantitative (Fig. 7.2a).
- Classification, y is categorical (Fig. 7.2b).
- Clustering, no y is provided - it is learned from dataset (Fig. 7.2c).
- Semi-supervised learning, combination of labeled and unlabeled data (Fig. 7.2d).
- Anomaly detection, somewhere between classification and clustering (Fig. 7.2e).
- Segmentation
- Simulation
- Signal processing tasks: noise removal, smoothing (filling missing values), event/condition detection.



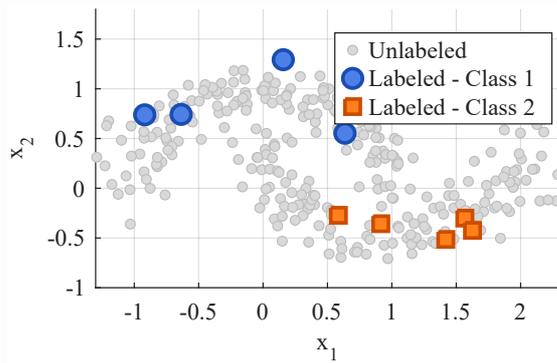
(a) Regression: what is the value of y for given x ?



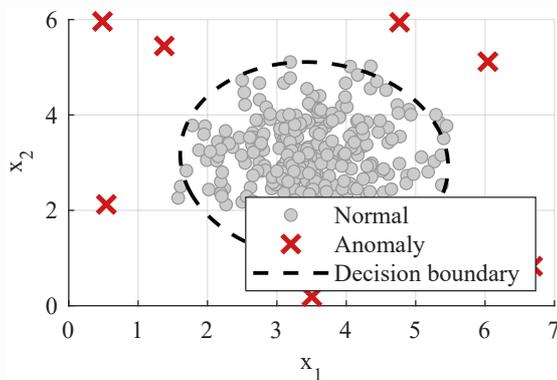
(b) Classification: what is the class of a new sample?



(c) Clustering: which samples belong together?



(d) Semi-supervised: few labeled samples guide learning from many unlabeled ones.



(e) Anomaly detection: which samples deviate from the normal pattern?

Figure 7.2: Examples of common ML goal types.

Note, there is possible to have two or more goals for the same dataset.

7.1.2 Model

We assume that there is an underling problem (e.g., regression and classification) formulation is of the form

$$y = h(\mathbf{x}) + \epsilon \quad (1)$$

where $h(\mathbf{x})$ is the true unknown function and ϵ is some irreducible noise. Sometimes, zero-mean noise is assumed. The values of \mathbf{x} (scalar or vector) and y are known (it is the dataset).

The goal is to find the function $f(\cdot; \mathbf{w})$ that approximates $h(\mathbf{x})$. The way to define the $f(\cdot; \mathbf{w})$ is termed *model* that depends on some model parameters vector \mathbf{w} . The process of finding parameters \mathbf{w} is called learning, such as the resulting model can provides predictions

$$\hat{y}_0 = f(\mathbf{x}_0; \mathbf{w}) \quad (2)$$

for some new data \mathbf{x}_0 .

Parameters vs hyper-parameters

There is a conceptual difference between parameters and hyper-parameters.

Model parameters: Model parameters are learned directly from a dataset.

Hyper-parameters: Model parameters that are not learned directly from a dataset are called hyper-parameters. They are learned in in-direct way during cross-validation process in the follow.

Hyper-parameter optimization Selecting the most appropriate hyper-parameters value is called hyper-parameter optimization.

Parametric vs non-parametric models

There are two main classes of models: parametric and non-parametric, summarized in Table 7.1.

Table 7.1: Comparison of parametric and non-parametric models.

Aspect	Parametric	Non-parametric
Dependence on number of parameters on dataset size	Fixed	Flexible
Interpretability	Yes	No
Underlying data assumptions	Yes	No
Risk	Underfitting due to rigid structure	Overfitting due to high flexibility
Dataset size	Smaller	Best for larger
Complexity	Often fast	Often complex
Examples	Linear regression	k-NN, trees

The modern trend is to bridge the gap between interpretable and non-parameteric modeling.

7.1.3 Loss Function

Loss (or cost) function is a function that relates between dataset outputs \mathbf{y} and model outputs $\hat{\mathbf{y}}$. The parameters \mathbf{w} are minimum of that function,

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) \quad (3)$$

The minimization of the loss function is also termed *training*.

Loss Function Minimization

Goal: Minimum of the loss function for a given model.

Closed-form solution A closed-form solution for \mathbf{w} is a solution that is based on basic mathematical functions. For example, a "normal equation" is a solution for linear regression/classification.

Local-minimum gradient-based iterative algorithms This family of algorithms is applicable only for convex (preferably strictly convex) loss functions. For example, gradient descent (GD) and its modifications (e.g., stochastic GD) are used to evaluate NN parameters. Another example is the Newton-Raphson algorithm.

- Some advanced algorithms under this category also employ (require) second-order derivative $\frac{\partial^2}{\partial \mathbf{w}} \mathcal{L}$ for faster convergence.
- If either derivative is not available as a closed-form expression, it is evaluated numerically.

Global optimizers The goal of global optimizers is to find a global minimum of non-convex function. These algorithms may be gradient-free, first-derivative or second-derivative. The complexity of these algorithms is significantly higher than the local optimizer and can be prohibitive for more than a few hundred variables in \mathbf{X} .

7.1.4 Metrics

Metrics are quantitative performance indicator $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ of the model that relate between \mathbf{y} and $\hat{\mathbf{y}}$. Sometimes, the minimum of the loss function is also a metric, e.g. mean squared error (MSE).

7.2 Project Steps

A widely adopted framework for ML/DL projects is the Cross-Industry Standard Process for Data Mining (CRISP-DM), which defines six iterative phases:

1. **Business understanding** – define the problem, success criteria, and project plan.

2. **Data understanding** – collect initial data and perform exploratory analysis.
3. **Data preparation** – clean, transform, and construct the final dataset.
4. **Modeling** – select and train candidate models.
5. **Evaluation** – assess model performance against business objectives.
6. **Deployment** – integrate the model into production.

The process is iterative: findings in any phase may require revisiting earlier phases.

Practical guidelines

- *Understanding the problem*: domain knowledge is essential.
 - Define performance metric and performance goal.
 - Evaluate human-level performance, if relevant.
- *Data collection and preparation*: typically the most time-consuming step.
 - Ensure representative distribution.
 - Identify outliers.
 - Compute basic statistics and create visualizations.
 - Verify sufficient dataset size.
- *Model engineering*:
 - Start with a standard, pre-implemented model.
 - Avoid overly complex models for the baseline.
- *Baseline implementation*:
 - Build an end-to-end pipeline: data → model → loss → metrics.
 - Debug and verify a sufficient level of performance.
- *Performance evaluation*:
 - Identify overfitting and underfitting (bias/variance analysis).
 - Investigate errors and validate dataset integrity.
- *Performance improvement*:
 - Hyper-parameter optimization.
 - Iteratively refine: revisit the problem, add data, or improve the model until sufficient performance is achieved.
- *Model deployment*:
 - Deploy the model for the target application.

7.2.1 MLOps

Machine Learning Operations (MLOps) extends DevOps principles to ML systems, bridging the gap between model development and production deployment.

- **Develop:** algorithm training and testing, ETL / data pipelines, continuous integration and deployment (CI/CD).
- **Operate:** continuous delivery, model inference, monitoring and management.