5705.25 - Hagfrøðilig læring H25

October 28, 2025

Contents

1 Introduction

2 Supervised Learning: A First Approach

2.1 From Traditional Programming to Machine Learning

In traditional computer science, a program is explicitly written to map inputs to outputs:

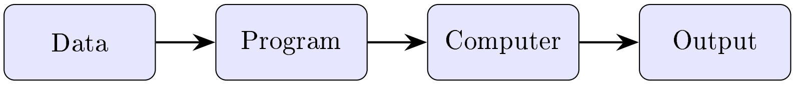
$$\text{Data} + \text{Program} \rightarrow \text{Computer} \rightarrow \text{Output}$$

Here, the program defines deterministic rules, such as `if-else` statements.

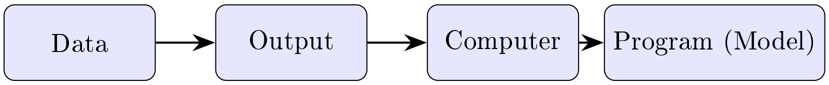
Machine learning inverts this paradigm:

$$\text{Data} + \text{Output} \rightarrow \text{Computer} \rightarrow \text{Program (Model)}$$

The computer now learns the mapping from examples rather than explicit programming. The more representative the data, the better the learned model.



Traditional Programming: Rules are explicitly coded.



Machine Learning: The computer learns the program from data.

2.2 Learning from Data

2.2.1 Example: ECG Data

Electrocardiogram (ECG) signals vary in time and across individuals. Each signal segment can be labeled as one of several heartbeat types:

$$y = \begin{cases} \text{NA} & \text{Normal Activity} \\ \text{AF} & \text{Atrial Fibrillation} \\ \text{RB} & \text{Resting Beat} \end{cases}$$

The task is to learn a model that maps input signals to the correct label.

2.2.2 Defining Learning

An **agent** learns from experience E with respect to a task T and a performance measure P if its performance on T, measured by P, improves with experience E.

2.3 Supervised Datasets

A supervised dataset consists of labeled examples:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

where x_i is a feature vector and y_i is the known output.

Examples:

- **Housing Prices:** y_i = price, x_i = location, size, year, etc.
- **Stock Prediction:** y_i = up/down, x_i = time-based features.

2.4 Types of Supervised Problems

- **Binary Classification:** $y \in \{0, 1\}$
- **Multiclass Classification:** $y \in \{1, 2, \dots, K\}$
- **Regression:** $y \in \mathbb{R}$

The goal is to find a function $h(x)$ that approximates y:

$$h(x_i) \approx y_i, \quad \forall i = 1, \dots, n$$

2.5 The Hypothesis Space

Let $\mathcal{H}$ denote the set of candidate functions (hypotheses). Each $h \in \mathcal{H}$ represents one possible model:

$$h(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

The objective is to select the hypothesis that best fits the data.

2.6 Loss Function and Model Selection

The **loss function** quantifies how well a hypothesis performs:

$$L_{\text{MSE}}(h, T) = \frac{1}{n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

$$L_{\text{MAE}}(h, T) = \frac{1}{n} \sum_{i=1}^{n} |h(x_i) - y_i|$$

A perfect fit ($L = 0$) may indicate overfitting — the model memorizes the training data and fails to generalize.

2.6.1 Expected Loss

We aim to minimize the expected loss:

$$\mathbb{E}_{(x,y)\sim P}[L(h(x), y)]$$

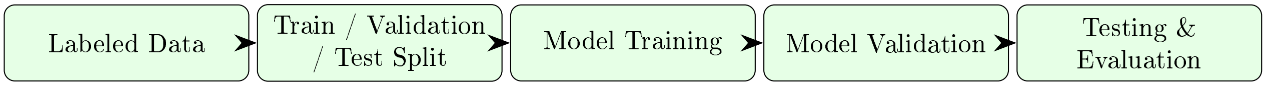
Since the population distribution P is unknown, we approximate it with train/validation/test splits:

$$80\% \text{ train}, \ 10\% \text{ validation}, \ 10\% \text{ test}.$$

2.6.2 No Free Lunch Theorem

No single model works for every problem — each dataset has its own structure.
Model choice must depend on the problem context.

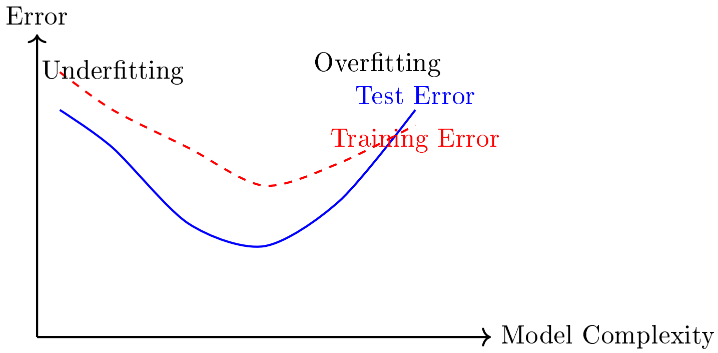
2.7 Supervised Learning Workflow



2.8 Underfitting and Overfitting

- **Underfitting:** Model too simple; fails to capture structure.
- **Overfitting:** Model too complex; captures noise.

Balancing bias and variance is crucial for generalization.



2.9 k-Nearest Neighbors (k-NN)

In k-NN, predictions are based on the labels of the k closest data points to a query x_*:

$$\|x_i - x_*\| = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{*j})^2}$$

Hyperparameter: k

- Low k: possible overfitting.
- High k: possible underfitting.

```
Input: Training set $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, query $x_*$, number of
       neighbors $k$
Output: Predicted label $\hat{y}_*$
foreach $(x_i, y_i) \in T$ do
    Compute distance $d_i = \|x_i - x_*\|$ (e.g., Euclidean)
end
Sort $d_i$ in ascending order;
Select $k$ nearest neighbors;
if classification then
    $\hat{y}_* \leftarrow$ majority label among $k$ neighbors
else
    $\hat{y}_* \leftarrow \frac{1}{k}\sum_{i=1}^{k} y_i$ (mean for regression)
end
```

Algorithm 1: k-Nearest Neighbors Algorithm

2.9.1 Normalization

Since distances depend on scale, normalize each feature:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

where μ_j and σ_j are the mean and standard deviation of feature j.

2.10 Summary

Supervised learning aims to infer a mapping from inputs to outputs based on labeled data. The process involves:

1. Defining the hypothesis space $\mathcal{H}$
2. Choosing a suitable loss function L
3. Optimizing $h \in \mathcal{H}$ to minimize L
4. Validating and testing to ensure generalization

A good model balances complexity and performance, avoiding both underfitting and overfitting.

3 Linear Regression