Course Of Machine Learning

By

Professor Abdellatif El Afia
ENSIAS- University in Rabat

أعوذ بالله من الشيطان الرجيم
بسم الله الرحمن الرحيم

وَإِذْ قَالَ رَبُّكَ لِلْمَلَٰٓئِكَةِ إِنِّي جَاعِلٌ فِي ٱلْأَرْضِ خَلِيفَةً ۖ قَالُوٓاْ أَتَجْعَلُ فِيهَا مَن يُفْسِدُ فِيهَا وَيَسْفِكُ ٱلدِّمَآءَ وَنَحْنُ نُسَبِّحُ بِحَمْدِكَ وَنُقَدِّسُ لَكَ ۖ قَالَ إِنِّيٓ أَعْلَمُ مَا لَا تَعْلَمُونَ

وَعَلَّمَ ءَادَمَ ٱلْأَسْمَآءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى ٱلْمَلَٰٓئِكَةِ فَقَالَ أَنۢبِـُٔونِى بِأَسْمَآءِ هَٰٓؤُلَآءِ إِن كُنتُمْ صَٰدِقِينَ

قَالُواْ سُبْحَٰنَكَ لَا عِلْمَ لَنَآ إِلَّا مَا عَلَّمْتَنَآ ۖ إِنَّكَ أَنتَ ٱلْعَلِيمُ ٱلْحَكِيمُ

قَالَ يَٰٓـَٔادَمُ أَنۢبِئْهُم بِأَسْمَآئِهِمْ ۖ فَلَمَّآ أَنۢبَأَهُم بِأَسْمَآئِهِمْ قَالَ أَلَمْ أَقُل لَّكُمْ إِنِّيٓ أَعْلَمُ غَيْبَ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ وَأَعْلَمُ مَا تُبْدُونَ وَمَا كُنتُمْ تَكْتُمُونَ

Example: Bank Credit

Develop a software (**machine**) able to take one of the following decisions (Task):

- Credit approval : Classification Task
- Amount of credit allocation: Regression Task
- Probability of credit approval : Probability Distribution Task

To build this machine, we should have a customer **historical data**.

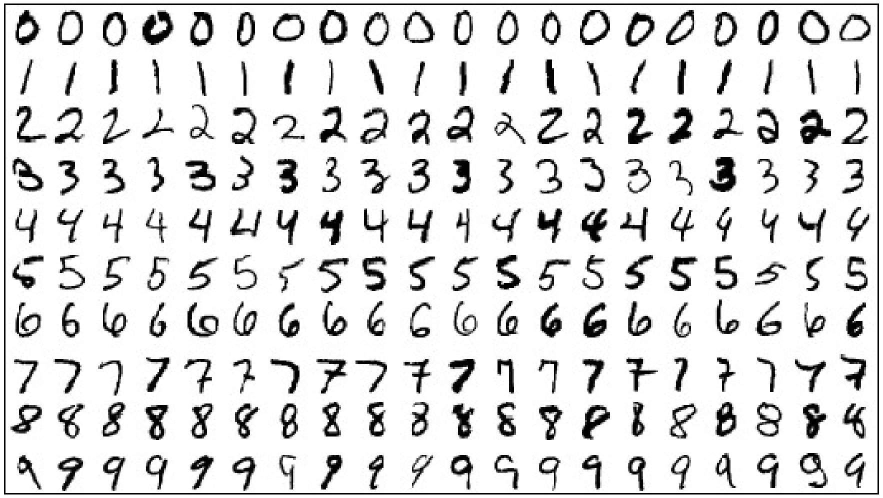
The objective is to let the machine **learn (Best Model in the Task's type chosen)** from the data.

➢ Machine learning.

Example: Digits Recognition

Develop a software (**machine**) able to recognize handwritten digits
Task: Multi-class Classification.

To build this machine, we should have a set of pictures **historical data**.



The objective is to let the machine **learn(Best Model)** from the data.

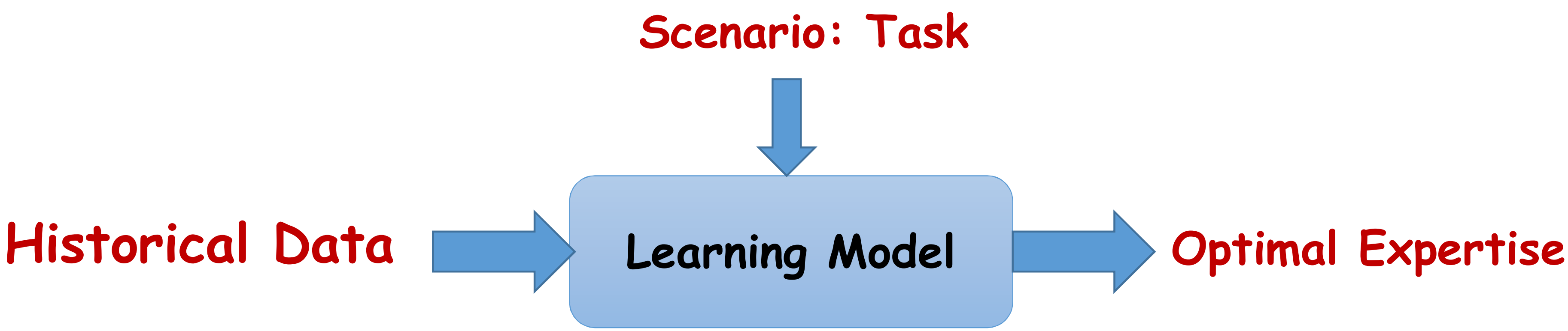
➢ Machine learning.

Overview of this Course

- Introduction to Machine Learning
- Part1: Machine Learning Theory
- Part2: Tasks Type: Classification’s Models – Regression’s Models, Multiclass Classification’models
- Part 3: Overfitting & Underfitting : How we improve the model’s learning

Introduction to Machine Learning

Definition of Machine Learning



Machine Learning is a process of **<u>experience (Historical Data)</u>** to gain **<u>expertise</u>**

- **Scenarios:** Supervised Learning - Unsupervised Learning - Reinforced Learning
- **Tasks:** Classification(Binary & Multi) - Regression - Probability Distribution Task

Optimal Expertise: h

Case Supervised Learning

- Historical Data: $\{(x_i, y_i)\}\ i \in I = \{1, \dots, m\}$,
- Purpose: it's to find the best function(Model), $\boldsymbol{h}$, such that
  - $h(x_i) = y_i \ \forall i \in I$
  - $h(x_i) = y_i \ \forall i \in F = \{m+1, \dots, \infty\}(\boldsymbol{Future})$
- Binary Classification Task :$y_i \in \{c_1, c_2\}$
- Multi classification Task: $y_i \in \{c_1, c_2, \dots \dots, c_{p-1}, c_p\}$
- Regression task: $y_i \in \mathbb{R}$
- Probability Distribution Task: $y_i \in [0,1]$

Using conditions of machine learning

The use of ML requires the fulfillment of three conditions:

1- Existence of a model to learn:

There is a correlation between input and output variables. We know that a model exists even if we do not know it.

2- Mathematical modeling is impossible:

We can not solve the model mathematically (no analytical solution).

3- Existence of data: (sufficient condition)

There is data that represents the model.

Machine learning scenarios

There are different learning scenarios to adapt with different situations and conditions.

The three main learning scenarios are:

- **Supervised learning(semi supervised learning)**
- **Unsupervised learning**
- **Reinforced learning**

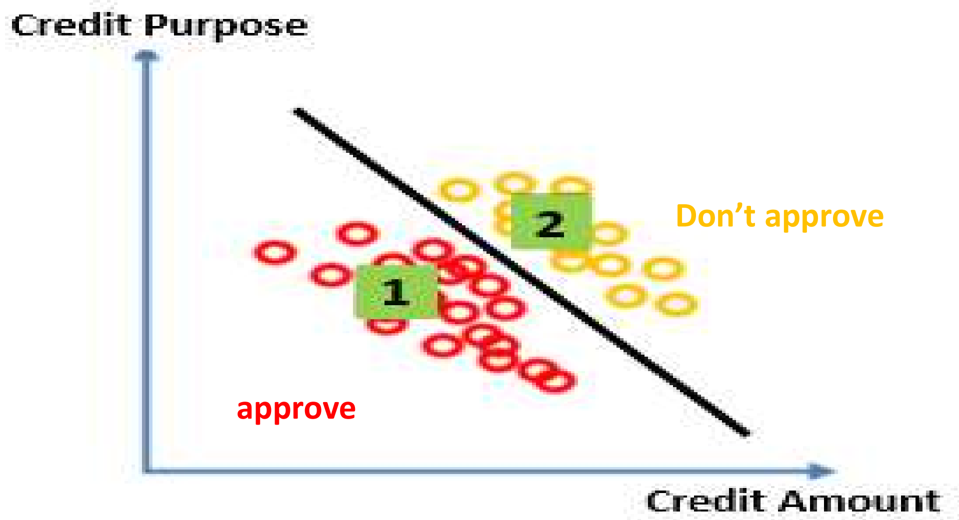
Machine learning scenarios

Supervised Learning

- The data form is: (**Inputs**, **Correct Outputs**)= (x_i, y_i)
- Learn from a dataset tagged by target variables.
- Classification, regression and ranking are tasks that belong to the scenario of supervised learning.

-Example:

Credit Approval: $y_i \in \{c_1, c_2\}$

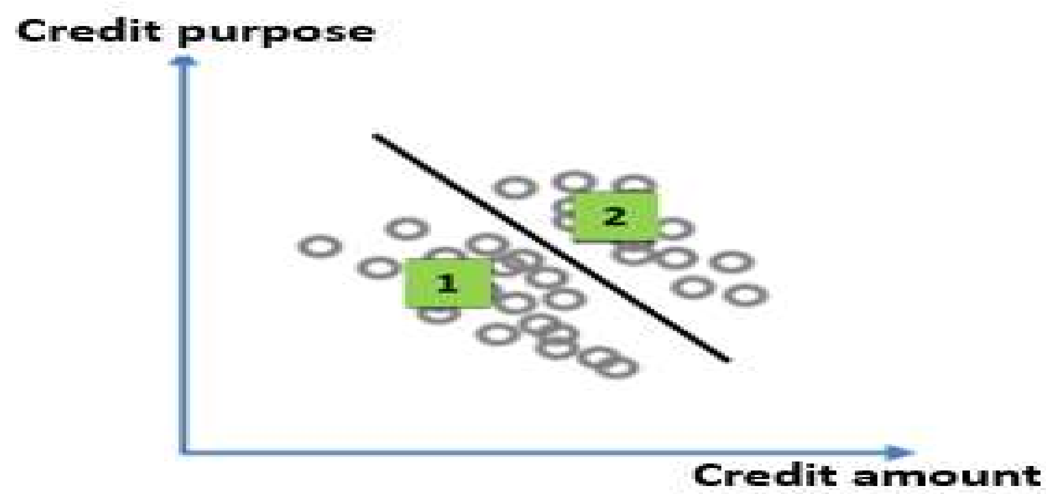


Machine learning scenarios

Unsupervised Learning

- The form of the data is: (Inputs)= $\left(\boldsymbol{x_i} \in \mathbb{R}^d\right)$ Such that $\boldsymbol{Features}$: $(\boldsymbol{x_i^1}, \dots, \boldsymbol{x_i^d})$
- Learn from a non-tagged dataset by target variables.
- Partitioning and dimension reduction are tasks that belong to the scenario of unsupervised learning.

Example: Customer clustering.



Machine learning scenarios

Reinforced Learning

- The data form is: (Input, Some Outputs, Reward for each output).= $\left(x_i, y_j, R(y)\right)$

- Learn by interaction with the environment and by observing the result of certain actions.

- It can be used for classification, regression tasks if the training data is insufficient.

Example: Child learning.

Types of data reception

- **Active reception of data:**

The learning Machine selects the data.

- **Passive reception of data:**

The user provides the data to the learning Machine. This form owns two types:

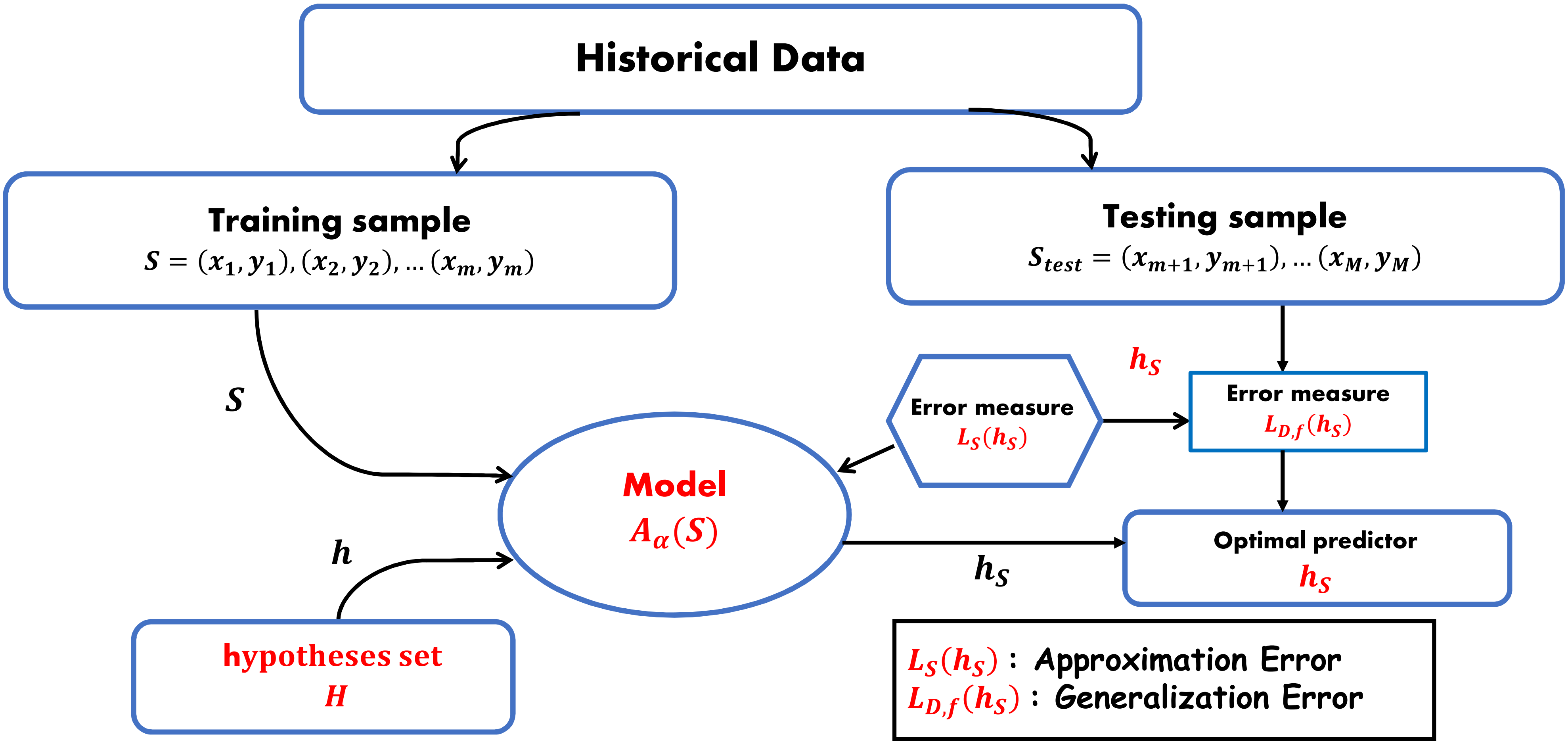
  - **Offline reception.**

The data is presented to the algorithm as a batch (all at once).

  - **Online reception.**

The data is presented to the algorithm incrementally (one by one).

Predictive Model: $\mathrm{M} = \{A_\alpha, H, L_S\}$ Passive - offline



Supervised Learning : Passive - offline

Task

Data
$S = \{(x_1, y_1), \dots, (x_m, y_m)\}$

Predictive Model
$\mathbf{M} = \{A_\alpha, H, L_S\}$

Optimal predictor
$\left(A_\alpha(S) \approx h_S \in \underset{h \in H}{\operatorname{argmin}}\{L_s(h)\}\right.$
$\left.L_{D,f}(h_S) \approx 0, f \in H, D : distribution\right)$

$x_i = (x_i^j, j = 1, \dots, d)$, x_i^j: feature

A_α: learn Model, H: set of hypotheses , L_S: empirical error function and α is vector of parameters. $h_S(x_i) = y_i$ $(h_S = \underset{h \in H}{argmin}\, L_s(h))$

It concerns the use of the best **features** x_i^j, to build the best **model** h_S by minimizing L_S in order to solve the best **tasks**. $L_{D,f}(h_S)$: Generalization Error

Data features

$$\boldsymbol{S} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), \dots, (\boldsymbol{x_m}, \boldsymbol{y_m})\}$$

- The features $\boldsymbol{x_i^j}$: $x_i = (x_i^j, j = 1, \dots, d)$ can take different forms:

- **Quantitative:** a real number.
- **Qualitative:** image, chain of letters, ...

- Labels $\boldsymbol{y_i}$ can take two forms:

- **Real values:**
  - $\boldsymbol{y_i} \in \mathbb{R} \Longrightarrow$ **Regression Task**
  - $\boldsymbol{y_i} \in [\mathbf{0}, \mathbf{1}] \Longrightarrow$ **Ranking Task**
- **Discrete values:** $\boldsymbol{y_i} \in \mathbb{N}$ **or** $\boldsymbol{y_i} \in \{\mathbf{0}, \mathbf{1}\} \Longrightarrow$ **Classification Task**

Supervised Learning Passive Offline Algorithm (SLPOA)

Goal: Find the Optimal Predictor

$x_i = \left(x_i^j, j = 1, \dots, d\right),\ \ y_i,$

x_i^j: **Feature**, f: **target function**

It consists on using the training sample to find the best hypothesis h_S that **Minimizes the** $L_S(h_S)$:

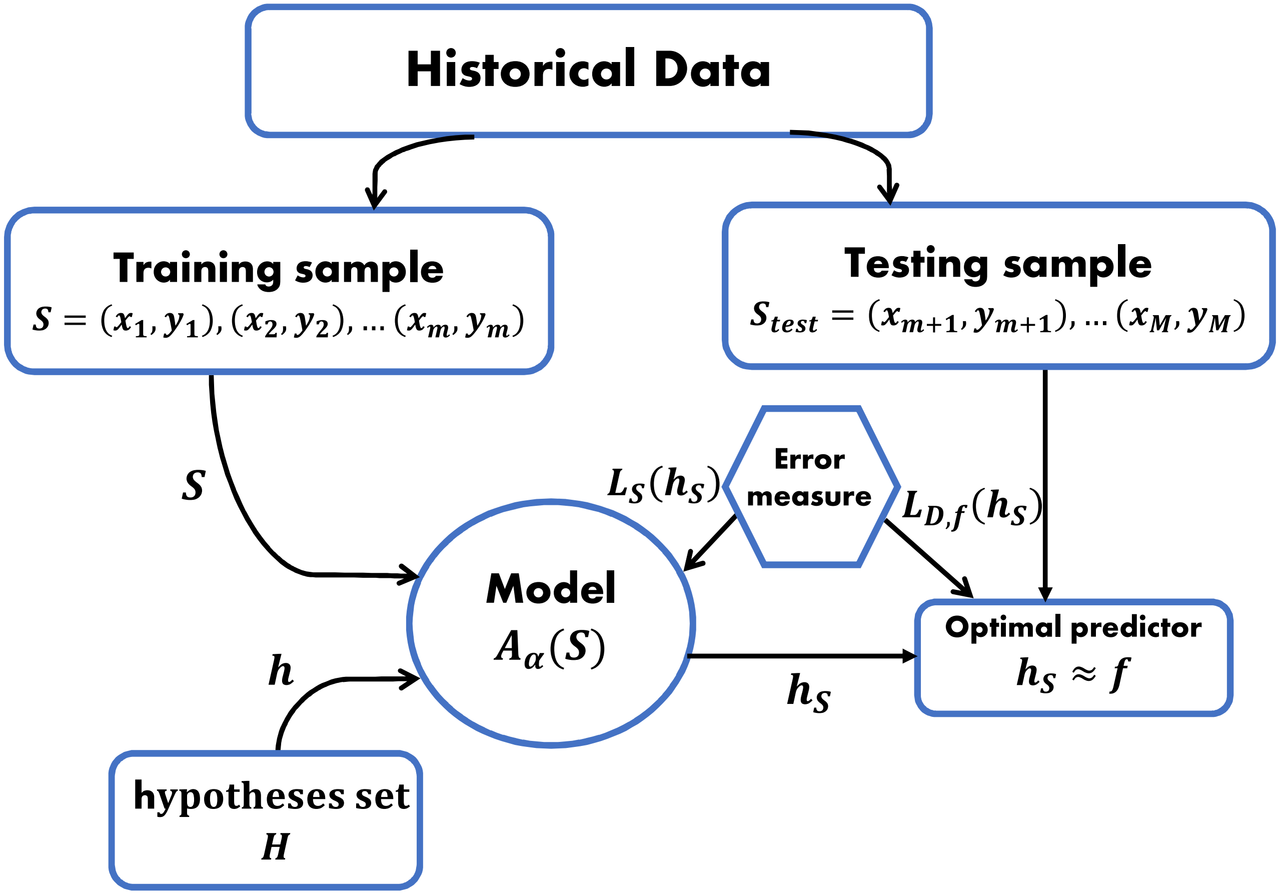
- **Approximation Error**
- **Empirical Error**
- **Loss Function**

Or Maximizes the **Approximation Capacity**

Then using the testing sample to measure the $L_{D,f}(h_S)$ of h_S:

- **Generalization Error**
- **Generalization Capacity**
- **D is a ditribution for measuring the quality of** h_S,

$L_{D,f}(h_S) = \mathrm{P}[\mathrm{h_S(x)} \neq \mathrm{f(x)}]\ \ \mathrm{P}{\sim}D$



Generalization Error: $L_{D,f}(h_S)$

- $L_{D,f}(h_S) = \mathbf{P}[\mathbf{x}, \mathbf{h_S}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x})] \in [\mathbf{0}, \mathbf{1}]$ $\mathbf{x} \in \boldsymbol{his} \cup \boldsymbol{Future}$

- D is a ditribution for measuring the quality of h_S , $\mathbf{P}[\mathbf{h_S}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x})]$ $\mathbf{P} \sim \boldsymbol{D}$

- $\mathbf{P}[\mathbf{x} \in \{\mathbf{Histo}\ \text{and}\ \mathbf{Future}\}\text{:}\ \mathbf{h_S}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x})] < \boldsymbol{\delta} \to \mathbf{0}$

$ERM: L_S(h_S)$, GE: $L_{D,f}(h_S)$

Classification:

h_S(ahmed)= yes or non, h_S is an hyperplan

- $L_S(h_S)$: mis***classified number****: training (Example: cours et TP) 18*
- $L_{D,f}(h_S)$>> $L_S(h_S)$? **overfitting**
- $L_{D,f}(h_S)$: **Generalization Error;: testing (Example: examen) 5**
- D**?**
- $Y = \{y_i\}$: label set

regression

- h_S(ahmed)= Amount of credit $\in \mathbb{R}$, $h_S$is a function : Linear, non-Linear

Ranking: distribution

h_S(ahmed)= probability$\in [0,1]$, = P(Credit Approval / ahmed)

Bank Credit: Credit Approval

Task: Binary Classification

Aim: Given the history $S \subset X \times Y$, find an optimal prediction model (separator) for future data.

Tool: Machine learning. $S = \{(x_1, y_1), \dots, (x_M, y_M)\} \in (X \times Y)^M$,

Inputs of the training algorithm:

- **Labels set:** $Y = \{0,1\} = \{Approved\ credit, non - approved\ credit\}$
- **Training set:** $S_T = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (X \times Y)^m$,
- **Testing set** $S_{Test} = \{(x_{m+1}, y_{m+1}), \dots, (x_M, y_M)\} \in (X \times Y)^{M-m}$
- **Feature set:** $x_i \in X$, $x_i = customers\ data, i = 1, .. M$

Features	Type
Present employment	Qualitative
Duration in month	Numerical
Credit history	Qualitative
Purpose	Qualitative
Age	Numerical
Number of existing credits at this bank	Numerical
Credit amount	Numerical

Outputs of the training algorithm:

- **Optimal hypothesis:** h = the best separator using S
- **Generalized model using** S_T

Bank Credit: Credit Allocation

Task: Regression

Aim: Given the history S, find an optimal prediction model (function) for future data.

Tool: Machine learning.

Inputs of the training algorithm:

- **Training set:** $S = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (X \times Y)^m$,
- **Testing set** $S_T = \{(x_{m+1}, y_{m+1}), \dots, (x_M, y_M)\} \in (X \times Y)^{M-m}$
- **Feature set:** $x_i \in X$, $x_i = customers\ data, i = 1, .. M$
- **Labels set:** $Y = \text{Amount of credit} \in \mathbb{R}$

Outputs of the training algorithm:

- **Optimal hypothesis:** h = the best function using S
- **Generalized model using** S_T

Features	Type
Present employment	Qualitative
Duration in month	Numerical
Credit history	Qualitative
Purpose	Qualitative
Age in years	Numerical
Number of existing credits at this bank	Numerical
Credit amount	Numerical

Bank Credit: Probability of Credit Approval

Task: Logistic Regression

Aim: Given the history S, find an optimal prediction model (probability of distribution $P(Y/X)$ because y_i are random and follow a binomial distribution) for future data.

Example: two customers with the same information but different credit approval decisions.

Tool: Machine learning.

Inputs of the training algorithm:

- **Training set:** $S = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (X \times Y)^m$,
- **Testing set** $S_T = \{(x_{m+1}, y_{m+1}), \dots, (x_M, y_M)\} \in (X \times Y)^{M-m}$
- **Feature set:** $x_i \in X$, $x_i = customers\ data, i = 1, .. M$
- **Labels set:** $Y = \text{probability of credit } approval \in [0,1]$

Outputs of the training algorithm:

- **Optimal hypothesis:** h = the best probability distribution using S
- **Generalized model using** S_T

Features	Type
Present employment	Qualitative
Duration in month	Numerical
Credit history	Qualitative
Purpose	Qualitative
Age in years	Numerical
Number of existing credits at this bank	Numerical
Credit amount	Numerical

Digits Recognition: Handwritten Digits Recognition

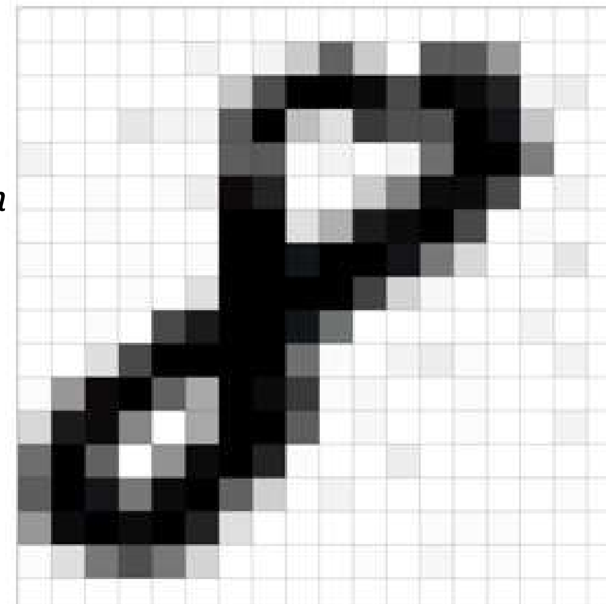
Task: Multi-class Classification

Aim: Given the history S, find an optimal prediction model (separator) for future data.

Inputs of the training algorithm:

- **Labels set:** $Y = \{0,1,2, \dots, 9\}$
- **Training set:** $S_T = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (X \times Y)^m$,
- **Testing set** $S_{Tes} = \{(x_{m+1}, y_{m+1}), \dots, (x_M, y_M)\} \in (X \times Y)^{M-m}$
- **Feature set:** number of pixels (18x18), $x_i \in \mathbb{R}^{324}$

$x_i = (x_i^1, x_i^2, \quad , x_i^{324})$



Outputs of the training algorithm:

- **Optimal hypothesis:** h = the best separator using S
- **Generalized model using** S_T

Models: Regression

Definition:

The objective of regression task is to find a function, in order to approximate real-valued targets.

Regression

Linear Regression

$$h(x) = \sum_{i=0}^{d} w_i x_i$$

$$x = (x_1, \dots, x_d)$$

$$w_{optimal} = \underset{w \in R^{d+1}}{\operatorname{argmin}} \; L_S(w)$$

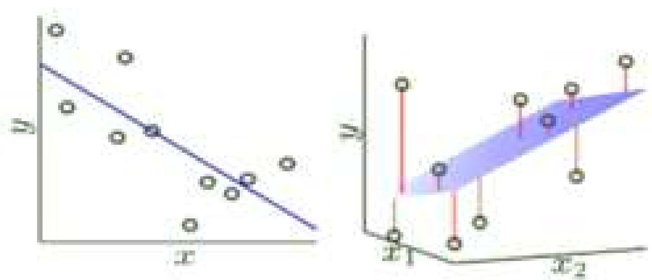
Non-Linear Regression

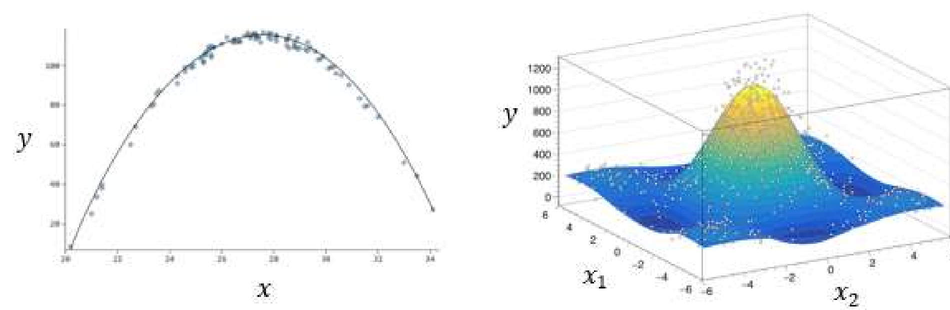
$$h(x) = \sum_{j=0}^{k} w_j x_1^j$$

$$x = (x_1)$$

(polynomial regression)

$$w_{optimal} = \underset{w \in R^{k+1}}{\operatorname{argmin}} \; L_S(w)$$



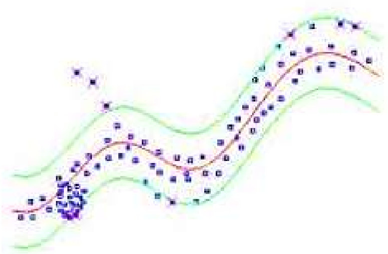


Models: Regression

Nonlinear models

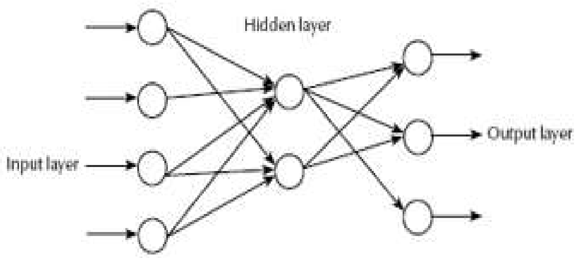
SVR-S4

Using the concept of the marge



MLP (ANN)- S4

Using the concept of the combination of many nonlinear functions



Models: Classification

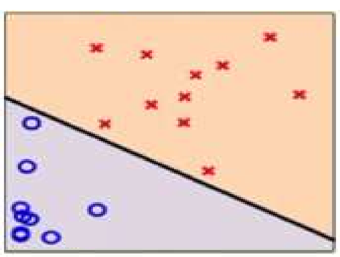
- Classification:

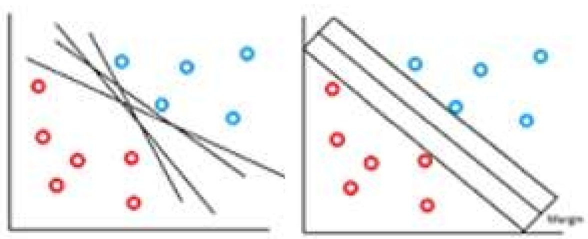
- Data type either:
  - **Linearly separable**
  - **Nonlinearly separable**

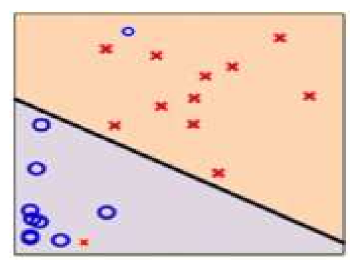
Models: Classification

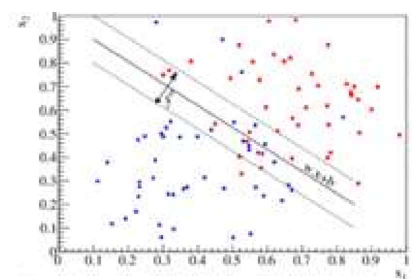
Linearly separable data

- **Without noise**
  - **Perceptron** → **PLA**

    
  - **Hard SVM -S4** → **Best separator**

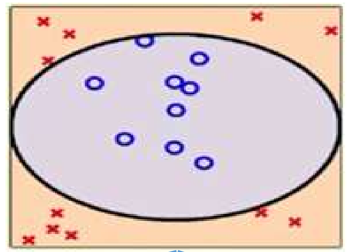
    
- **With noise**
  - **Adaline** → **Widrow-Hoff**
  - **Perceptron** → **Pocket**

    
  - **Soft SVM -S4** → **Best separator**

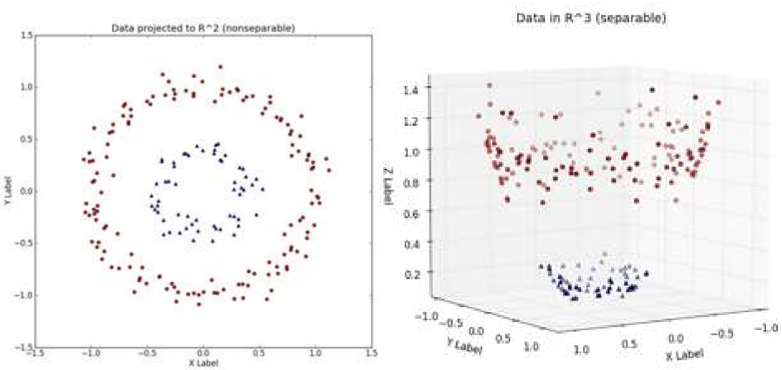
Models: Classification

Nonlinearly separable data



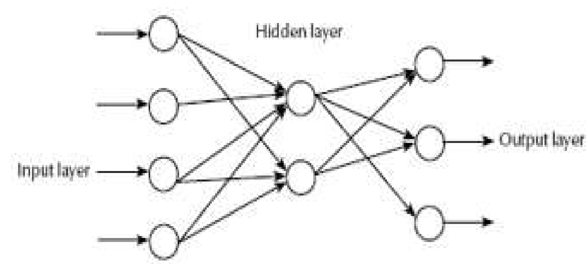
Nonlinear Transformation

Linear model

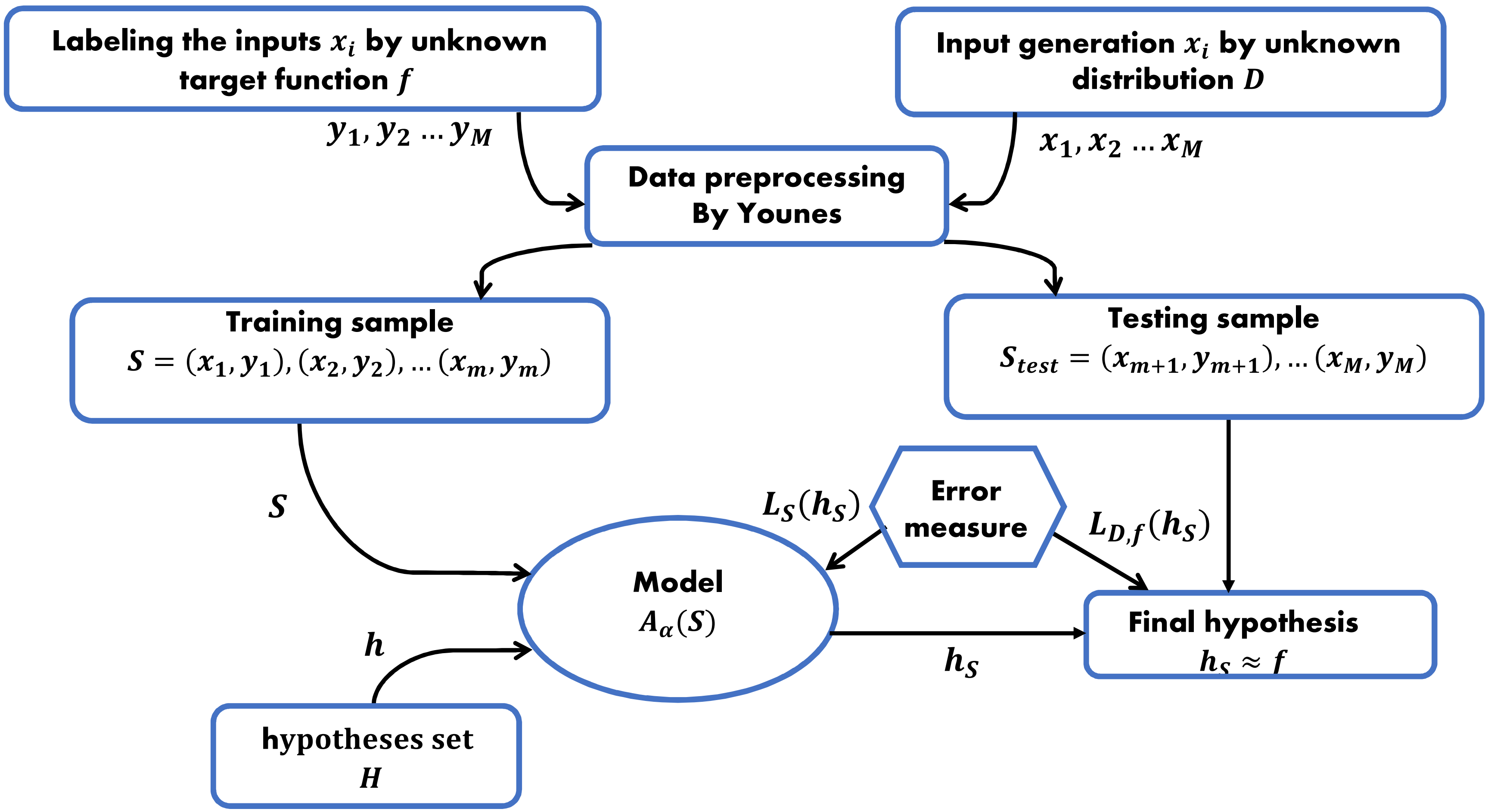


Nonlinear models- S4

MLP



Supervised Learning Passive Offline Algorithm (SLPOA)



Training Process(mahraoui)

Tasks (classification or regression)

Training Data
$S = \{(x_1, y_1), \dots, (x_m, y_m)\}$

Predictive Model
$\mathbf{M} = \{A_\alpha, H, L_S\}$

Approximation Capability
$(A_\alpha(S) = h_S \in \underset{h \in H}{\operatorname{argmin}}\{L_s(h)\}$

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(y_i - h(x_i))^2 \textit{ (regression case)}$$

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[h(x_i) \neq y_i]} \textit{ (classification case)}$$

$L_S(h)$: Empirical error function ; $x_i = (x_i^j, j = 1, \dots, d)$, x_i^j: Feature

A_α: Model, H: set of hypotheses , and α is vector of parameters.

It concerns the use of the best **features** x_i^j, to build the best **model** h_S by minimizing L_S in order to solve the best **tasks**.

Testing process

Testing data

$S_{test} = \{(x_{m+1}, y_{m+1}), \dots, (x_M, y_M)\}$

Optimal hypothesis h_S

Generalization Capability

$L_{D,f}(h_S)$

Such that:

$x_i = (x_i^j, j = 1, \dots, d)$, x_i^j: features

S_{test}: test set, h_S: optimale hypothesis. $\mathcal{D}$: Probability Distibution, $L_{D,f}$: general error.

$L_{\mathcal{D},f}(h) \stackrel{\text{def}}{=} \underset{x\sim\mathcal{D}}{\mathrm{P}}[x, h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}(\{x: h(x) \neq f(x)\})$: General error.

$$L_{D,f}(h_S) \approx L_{test}(h_S) = \frac{\sum_{i=m+1}^{M}(y_i - h_S(x_i))^2}{M-m} \quad \textit{(regression case)}$$

$$L_{D,f}(h_S) \approx L_{test}(h_S) = \frac{\sum_{i=m+1}^{M} \mathbb{1}_{[y_i \neq h_S(x_i)]}}{M-m} \quad \textit{(classification case)}$$

Overfitting-underfitting

- $L_S(h)$: empirique error (approximation capability).
- $L_{D,f}(h)$: general error (generalization capability).

Overfitting If $\boldsymbol{L_{D,f}(h) \gg L_s(h)}$

We say that the algorithm has a poor generalization capacity.

Underfitting if $\boldsymbol{L_s(h) \gg 0}$

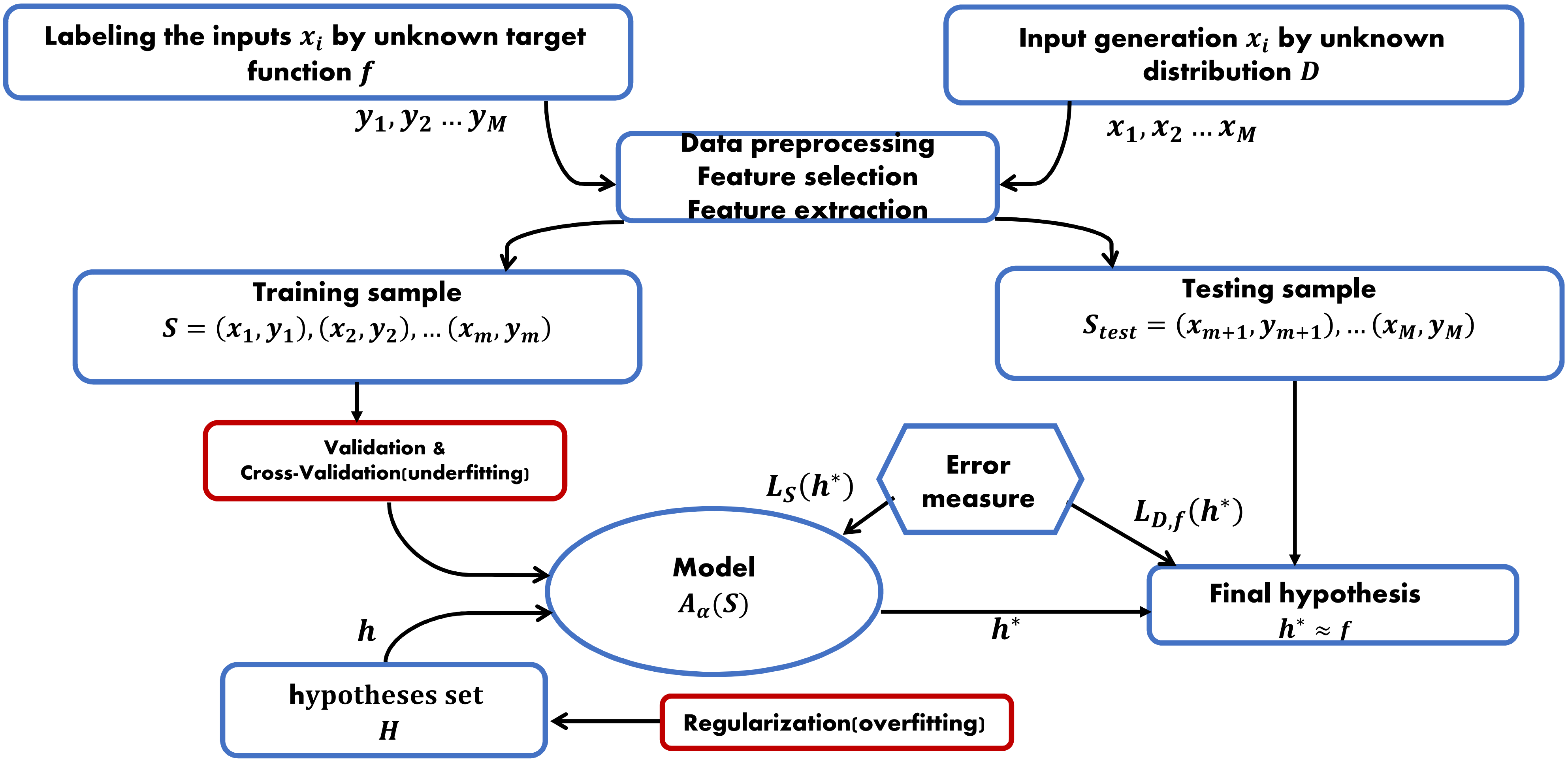
We say that the algorithm has a poor aproximation capacity

To remedy those problems, the following techniques are used:

Regularization: imposing a constraint **Overfitting**

Validation & Cross validation: selection of the best **$\boldsymbol{\alpha}$, or the best model**

Supervised Learning Passive Offline Algorithm (SLPOA)



Objective

What is learning?

- PAC learning.

How can a machine learn?

- ERM.

Is data learnable? S

- Uniform convergence.

What is the amount of data needed for learning?

- Uniform convergence.

How learning might fail?

- No-free-lunch theorem.

How can we measure the complexity of a model?

- VC dimension and Covering number.

Is the model a good learner?

- Tradeoff Bias/Variance. estimation of $\boldsymbol{L_{D,f}}$

How can we improve the model's learning?

- Regularization and Validation & cross-validation.

Outline

Part 1: Machine learning theory

- Discover the basic concepts of ML.
- Study the relationship between $L_S(h)$ and $L_{D,f}(h)$.
- Study the probability distribution of the data D.
- Study the labeling function f.
- Study the hypothesis set H.
- Study the model A_α.
- Find the best approximation of $L_{D,f}(h)$.

Part 2: Classification and regression Models

- Discover classification models.
- Implement classification models.
- Discover regression models.
- Implement regression models.
- Study nonlinear transformations.

Part 3 Regularization, Validation and Cross-Validation

- Fight against the overfitting and Underfitting.

Outline

Part 1 Machine Learning Theory Course

Learning framework

Uniform convergence

Learnability of infinite size hypotheses classes

Tradeoff Bias/Variance

Part 2: Learning Models TP

Classification

Regression

Part 3: Overfitting **course and TP**

Kernel

Regularization, Validation et Cross-Validation

Feature Selection

References

- Abdellatif El Afia: Machine Learning from Theory to Algorithms

Youtube: https://www.youtube.com/watch?v=lS3FAlCiuTs&t=531s

- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin. Learning from data.

Youtube: https://www.youtube.com/watch?v=mbyG85GZ0PI&list=PLnIDYuXHkit4LcWjDe0EwlE57WiGlBs08

- Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms.

Youtube: https://www.youtube.com/watch?v=b5NlRg8SjZg&list=PLFze15KrfxbH8SE4FgOHpMSY1h5HiRLMm

Livre: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf