

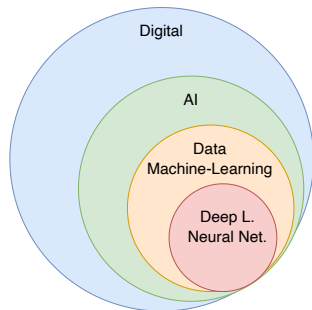
# INTRODUCTION À NUMPY (ET MATPLOTLIB+PANDAS)

Vincent Guigue  
[vincent.guigue@agroparistech.fr](mailto:vincent.guigue@agroparistech.fr)





# Artificial Intelligence & Machine Learning



Input ( $\mathbf{x}$ )	Output ( $\mathbf{Y}$ )	Application
email	→ spam? (0/1)	spam filtering
audio	→ text transcript	speech recognition
English	→ Chinese	machine translation
ad, user info	→ click? (0/1)	online advertising
image, radar info	→ position of other cars	self-driving car
image of phone	→ defect? (0/1)	visual inspection

**AI:** computer programs that engage in tasks which are, for now, performed more satisfactorily by human beings because they require high-level mental processes.

*Marvin Lee Minsky, 1956*

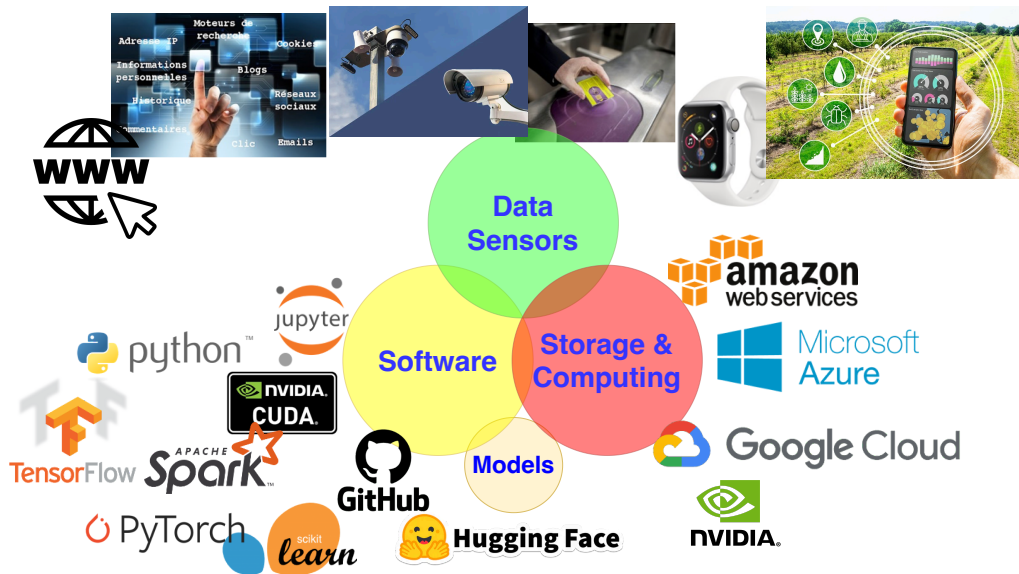
**N-AI (Narrow Artificial Intelligence)**, dedicated to a single task

**≠ G-AI (General AI)**, which replaces humans in complex systems.

*Andrew Ng, 2015*



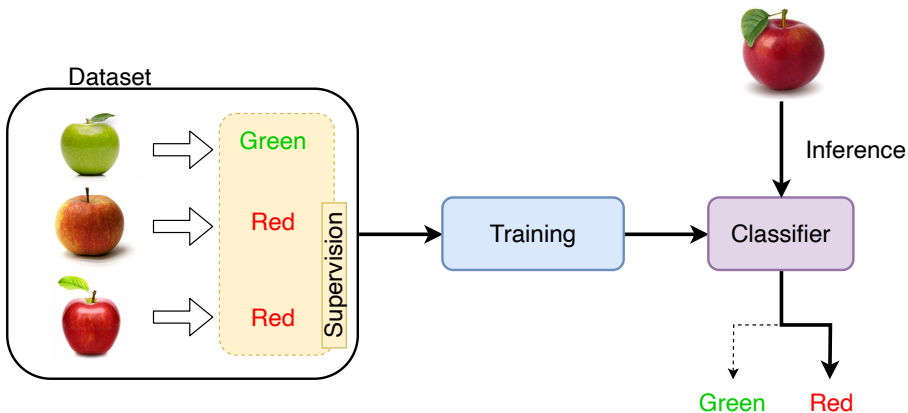
# The Ingredients of Artificial Intelligence





# Machine Learning Definition

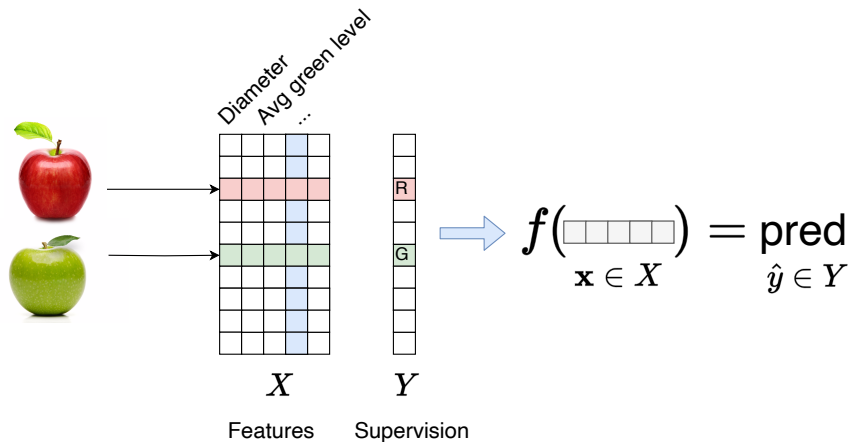
- 1 Collecting labeled **dataset**
- 2 Training **classifier**
- 3 Exploiting the model





# Machine Learning Definition

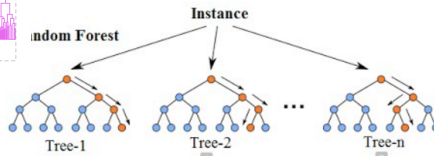
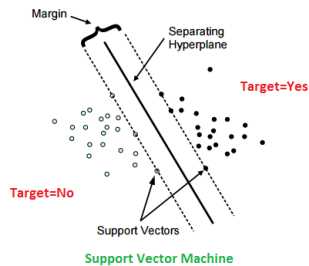
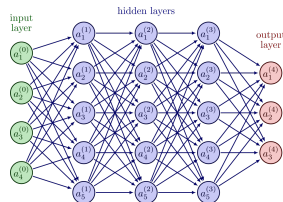
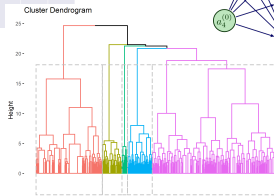
- 1 Collecting labeled **dataset**
- 2 Training **classifier**
- 3 Exploiting the model





# Machine Learning Definition

- 1 Collecting labeled **dataset**
- 2 Training **classifier**
- 3 Exploiting the model

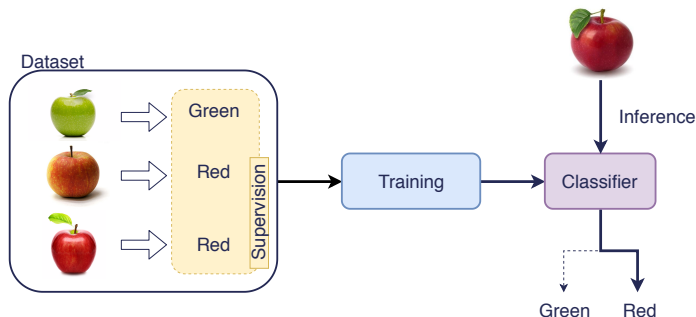




# Programmation *orientée données*

## ■ Python : langage unificateur (codage vs wrapper)

- *Calcul scientifique* : numpy
- *Machine-learning*: scikit-learn, pandas, matplotlib
- *Deep-learning*: pytorch
- Environnement de développement: Visual Studio Code / jupyter-notebook



Où se trouve les leviers de performance?

Dans les modèles...  
Mais surtout dans les chaînes de traitements !

# ORGANISATION





# Organisation (optim-iste/ale) du semestre

## ■ 5 séances - Machine Learning

- 1.5 séances - numpy = Mise à niveau en python, numpy, matplotlib
- 1 séance - scikit-learn = outils de base de pour la régression et la classification + évaluation robuste
- 1 séance - chaine de traitement, sélection de variables et pre-processing
- 1 séance - visualisation des données et optimisation des hyperparamètres
- 0.5 séance - Support projet en machine learning

## ■ 8 séances - Deep Learning en pytorch

- 1 séance - Introduction à pytorch, structure de données et gradient
- 1 séance - Perceptron & réseau de neurones
- 1 séance - Convolutional Neural Network & application en image
- 1 séance - Apprentissage de représentation (Embedding) & systèmes de recommandation
- 1 séance - Réseaux de neurones récurrents (RNN)
- 1 séance - Calcul d'attention (pour les RNN)
- 1 séance - Architecture Transformer
- 1 séance - Projet



# Numpy

3 séries de slides + notebooks



1 séance et demi

- 1 Python & numpy
- 2 Classification bayésienne
- 3 Algorithme(s) de gradient

⇒ On ne va pas tout faire! Mais on peut expliquer les idées

⇒ Chacun doit en tirer un message personnel optimal :-)



# Conclusion : passer à un nouveau langage...

## ■ **Cout faible**

- une fois que vous avez compris la logique générale

## ■ **Cout non négligeable:**

- Comprendre les forces et les faiblesses du langage
  - ... Et des environnements de développement
- Adapter sa manière de programmer (e.g. calculer un décile)
- Reprendre les bons réflexes (=aller vite)

⇒ [https://github.com/vguigue/tuto\\_numpy](https://github.com/vguigue/tuto_numpy)