

UC 2A - LE MACHINE LEARNING EN PRATIQUE

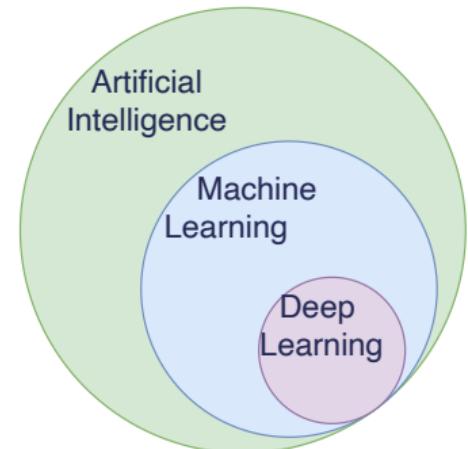
INTRODUCTION AUX OUTILS (NUMPY, SCIKIT-LEARN...)

Vincent Guigue, Romain Thoreau
vincent.guigue@agroparistech.fr



Intelligence Artificielle, Machine Learning et Programmation

| Input (X) | Output (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 |



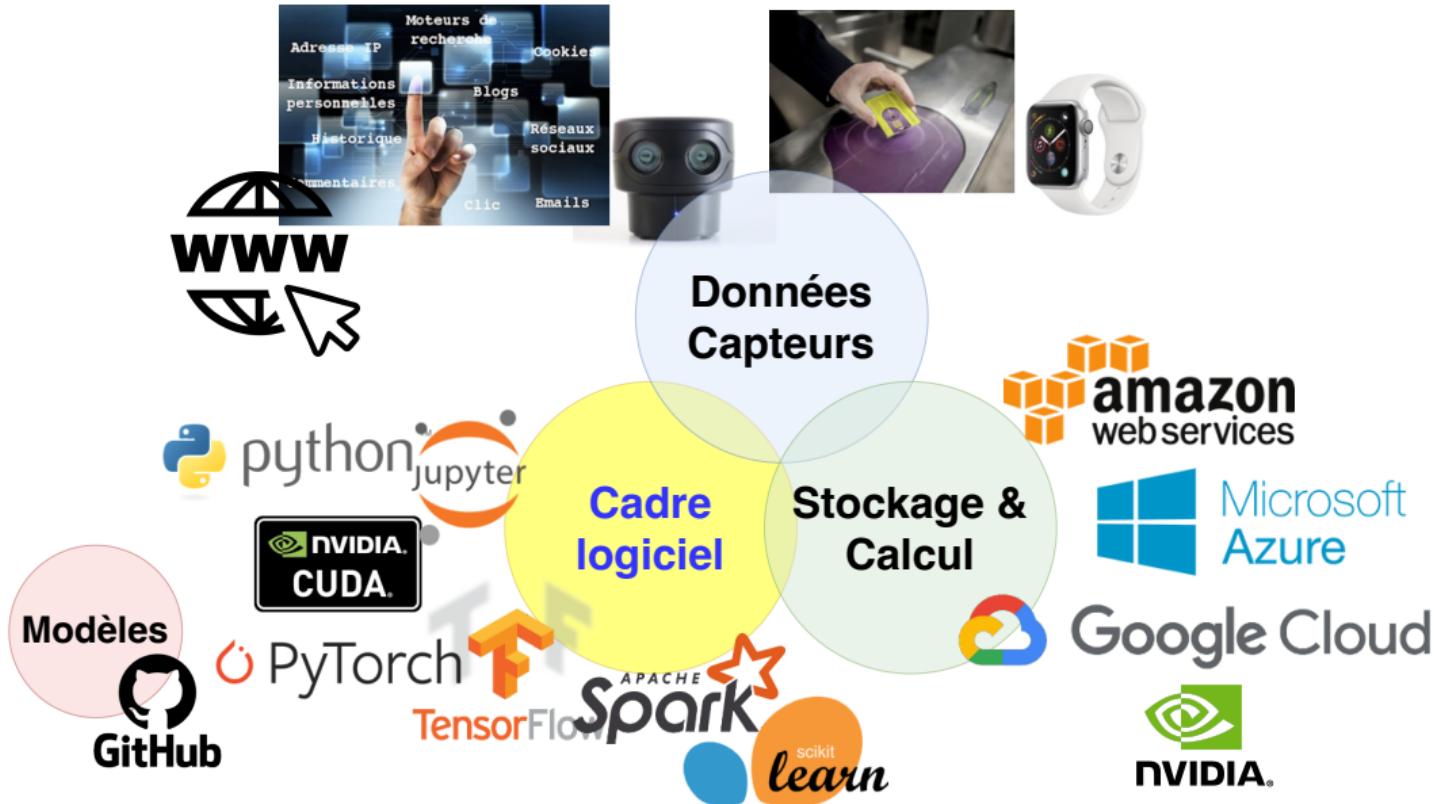
IA : programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau.

Marvin Lee Minsky, 1956

N-AI (Narrow Artificial Intelligence), dédiée à une tâche
≠ G-AI (General AI) qui remplace l'humain dans des systèmes complexes.

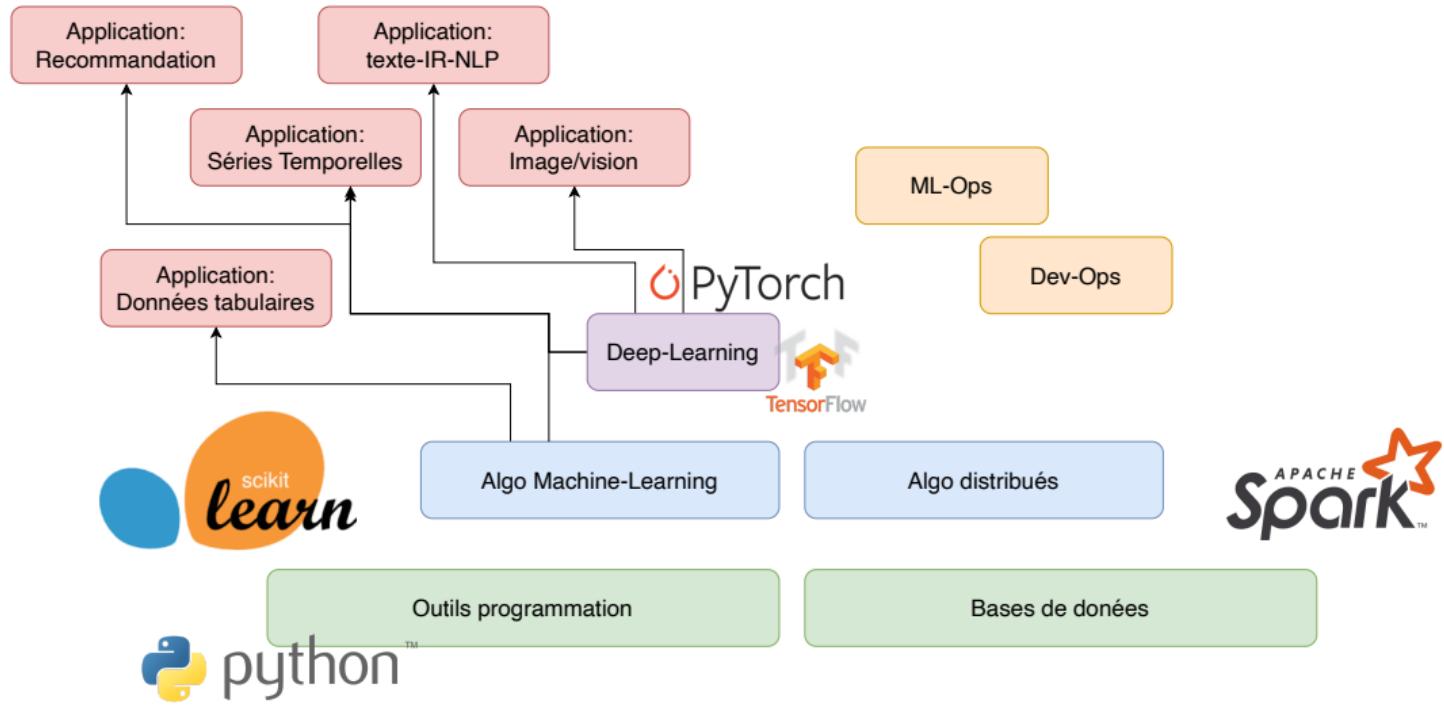
Andrew Ng, 2015

Ingrédients de l'Intelligence Artificielle



Enseignement de l'IA

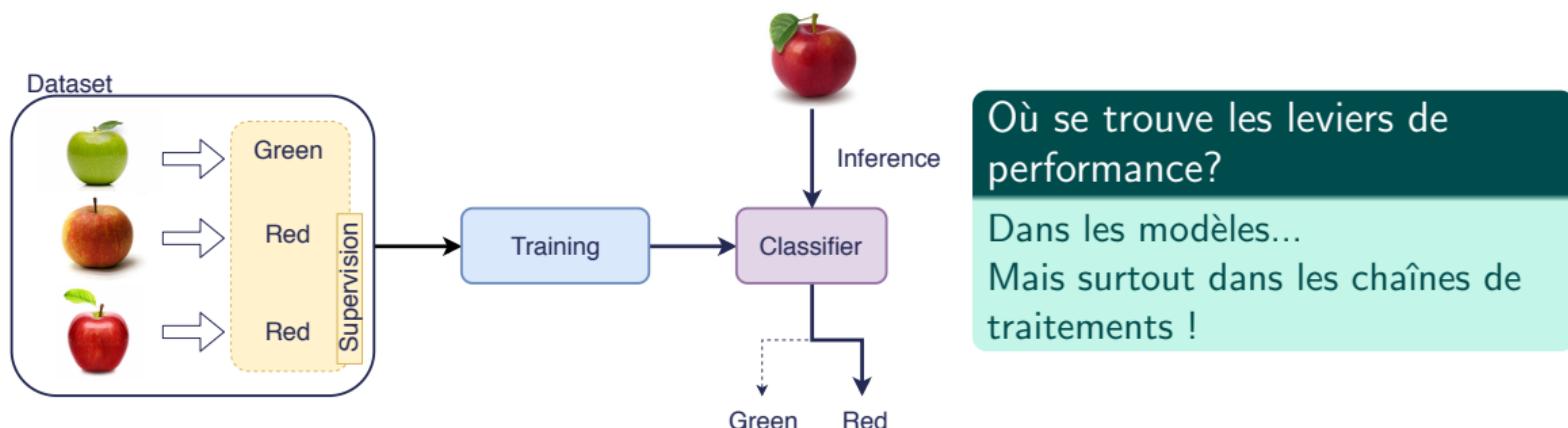
- Différents niveaux d'accès
- Différentes branches: types d'outils, application thématiques, ...



Programmation orientée données

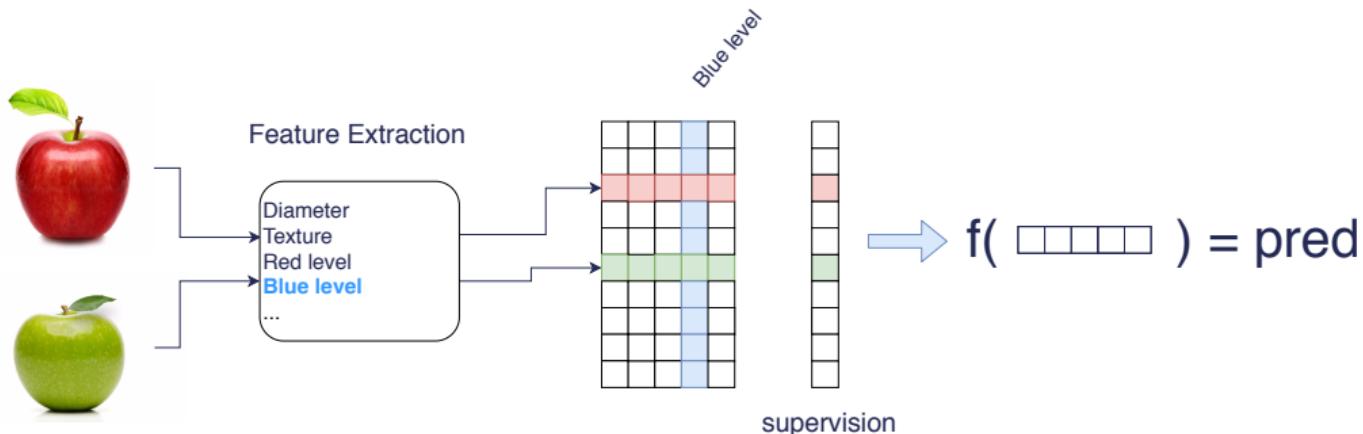
■ Python : langage unificateur (codage vs wrapper)

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



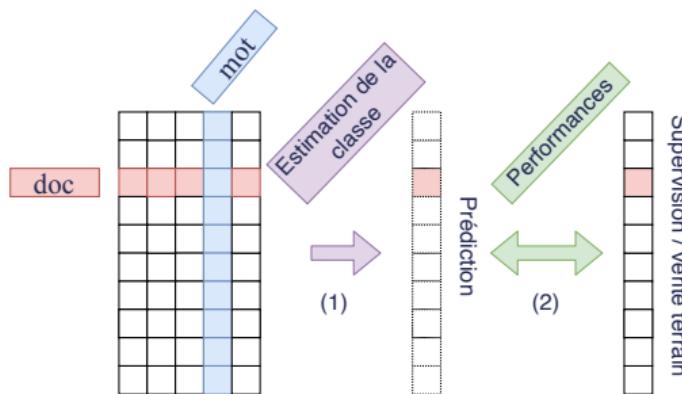
Programmation orientée données

- **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



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



ORGANISATION

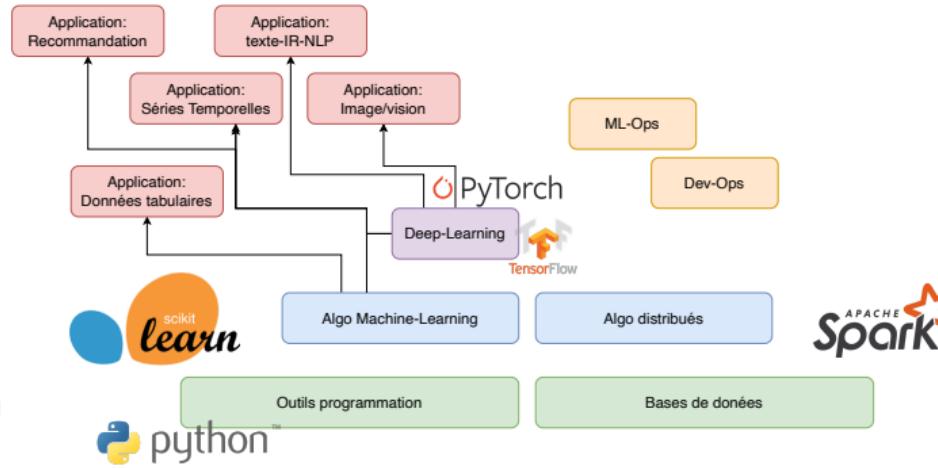
Organisation

■ 2 séances - Numpy (=5h)

- Mise à niveau en python, numpy, matplotlib
- Naïve Bayes (à la main)

■ 6 séances - Scikit-Learn (=6x3h30)

- Classificateurs Scikit-Learn : syntaxe, possibilités offertes
- Apprentissage supervisé et non-supervisé
- Évaluation
- Chaîne de traitements, sélection de modèles (grid-search...)
- Visualisation & post-traitements



■ 6 séances - Bouclage & projet (≈5.5x3h30)

Jupyter Notebook

■ Du code dans un navigateur web????

- Principe de textes à trous
- Bel outil pédagogique...
- ... avec des risques (contemplation)

et des limites (organisation de code sous-optimale)



A screenshot of a Jupyter Notebook interface. The title bar says "jupyter covid_19_dashboard Last Checkpoint: Last Friday at 11:45 PM (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar has icons for New, Open, Save, Run, Cell, Kernel, Help, Code, and Voila. On the right, there's a user icon, "Trusted", and "Python 3". Below the toolbar, there are two code cells:

```
In [13]: # importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_deaths.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_confirmed.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_recovered.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```

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 reflexes (=aller vite)

⇒ Devenir *data-scientist* n'a jamais été aussi facile... Mais il reste quelques savoir-faire et quelques pièges à éviter!