



Cours 1 - Introduction à l'Intelligence Artificielle

Alice Cohen-Hadria

Maîtresse de conférences à Sorbonne Université
alice.cohenhadria@gmail.com

Organisation du cours

- 5 cours
- 4 TDs (2h)
- 6 TPs (2h)



Alice
Cohen-Hadria



Sylvain Argentieri



Yann
Teytaut

Langage support en TP : Python

- 2 examens écrit : ER1 20% ER2 50%
- Contrôle de TP (évaluation du rendu après chaque TP) 30%

Organisation du cours - Plan

1. **Données, visualisation**

- a. Indicateurs statistiques
- b. Visualisation
- c. Implémentation en Python

2. **Réduction de dimension**

- a. Matrice de covariance
- b. Analyse en Composantes Principales

3. **Classification**

- a. K plus proches voisins
- b. Réseaux de neurones

Organisation du cours - Plan

1. Données, visualisation

- a. Indicateurs statistiques
- b. Visualisation
- c. Implémentation en Python

2. Réduction de dimension

- a. Matrice de covariance
- b. Analyse en Composantes Principales

3. Classification

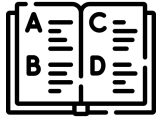
- a. K plus proches voisins
- b. Réseaux de neurones

Introduction



Intelligence artificielle - DéfinitionS

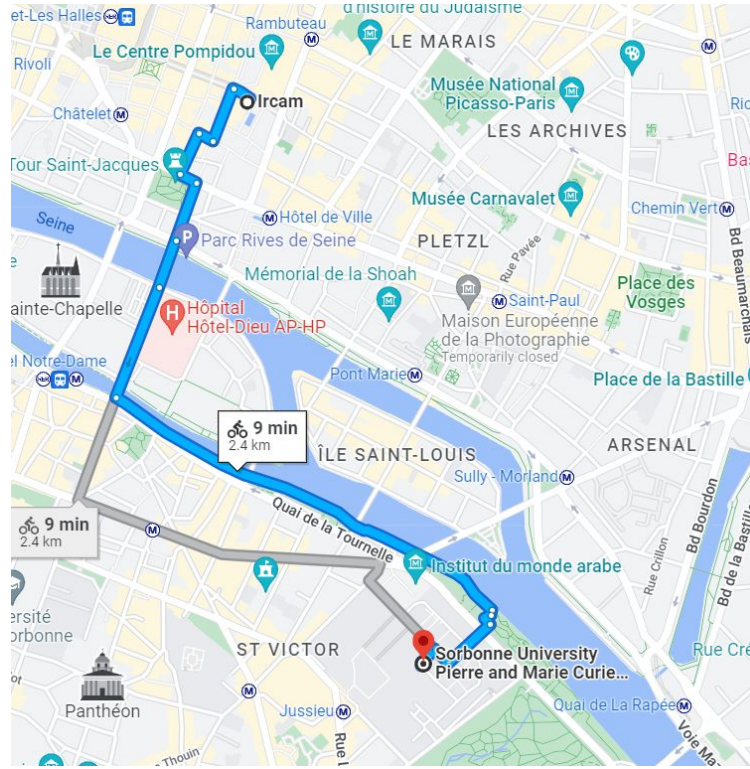
L'intelligence artificielle (IA) est « l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine » (définition Larousse).



Marvin Minsky (chercheur en IA) :

La construction de **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 tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique

Intelligence artificielle - Qu'est-ce qui est de l'intelligence artificielle ?



Intelligence artificielle - Qu'est-ce qui est de l'intelligence artificielle ?



Intelligence artificielle - Qu'est-ce qui est de l'intelligence artificielle ?



Mix Soul

Mix Soul

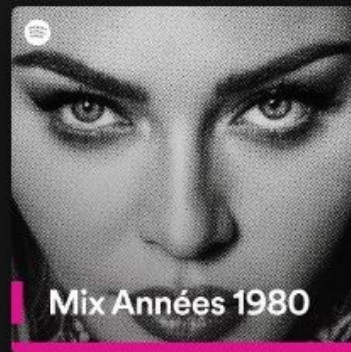
Marvin Gaye, The Jackson 5, Al Green et...



Mix Queen

Mix Queen

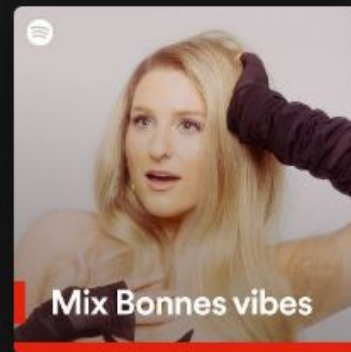
Billy Joel, Elton John, The Beatles et bien plus



Mix Années 1980

Mix Années 1980

Madonna, Whitesnake, Diana Ross et bien plus

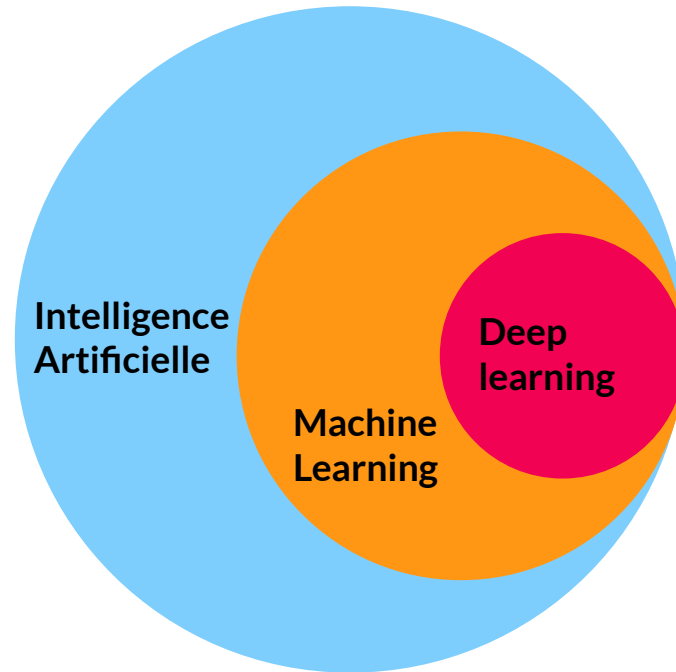


Mix Bonnes vibes

Mix Bonnes vibes

Meghan Trainor, Elton John, Aretha Franklin et...

IA ML DL



IA Introduction

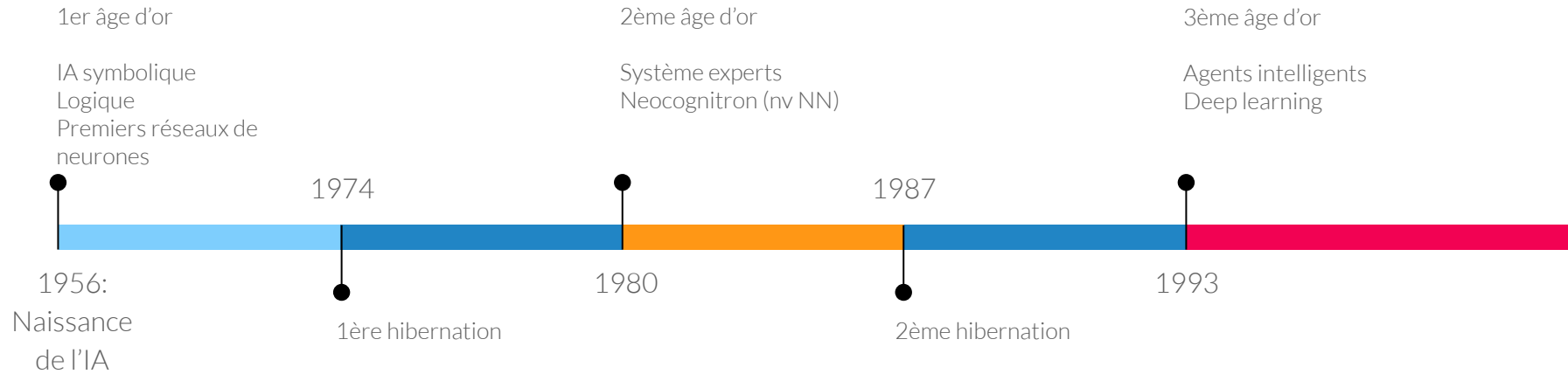
On définit une **tâche** que l'on veut résoudre (souvent difficile par d'autres moyens que l'IA ou non sensible).

Ensuite il faut choisir un **modèle** (une architecture de réseau de neurones par exemple).

Il nous faut ensuite des **données** pour **entraîner** le modèle choisi.

Les mots utilisés peuvent induire en erreur !

Un peu d'histoire ...



Données

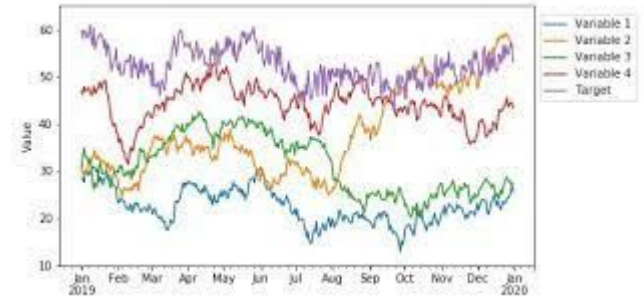
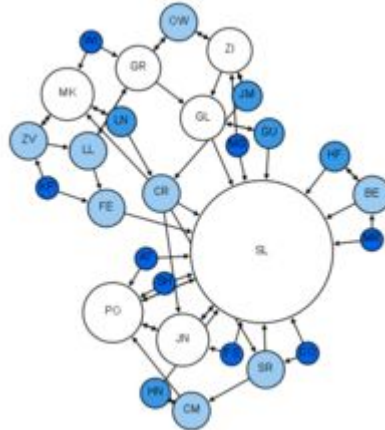
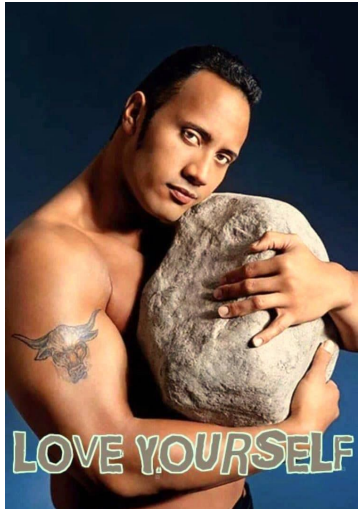


Données ??

Quels types de données on peut avoir ?

Données ??

	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0



Données ??

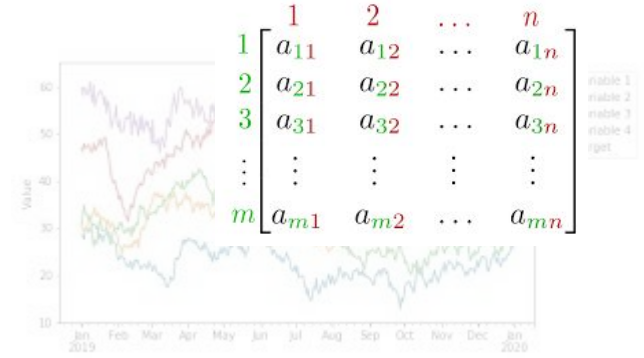
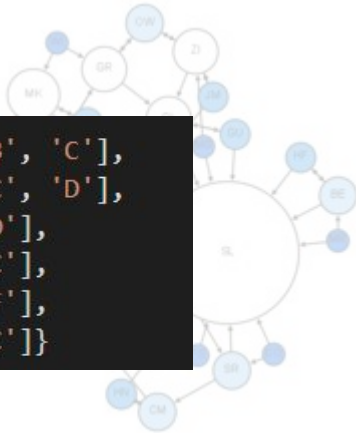


	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

```
csv = pd.read_csv("sample_data/donnee_pression.csv")  
csv.sample()
```



```
graph = {'A': ['B', 'C'],  
        'B': ['C', 'D'],  
        'C': ['D'],  
        'D': ['C'],  
        'E': ['F'],  
        'F': ['C']}
```



Données

On collecte, dans un hôpital, des données sur des patients:

- Leur âge
- Leur taille
- Leur poids
- Leur genre
- Leur pression artérielle
- Leur glycémie
- ...

Chacune de ces informations est un nombre (leur genre peut être converti en 0 ou 1 par exemple). En agrégeant ces données sur un patient, on obtient un vecteur.

Données

âge	16
taille	160
poids	59.0
genre	0
BP	110
glycémie	4.6



Patient 1

Données

âge	16
taille	160
poids	59.0
genre	0
BP	110
glycémie	4.6

Patient 1

âge	67
taille	170
poids	73.9
genre	1
BP	122
glycémie	7.0

Patient 2

Etc...

Données

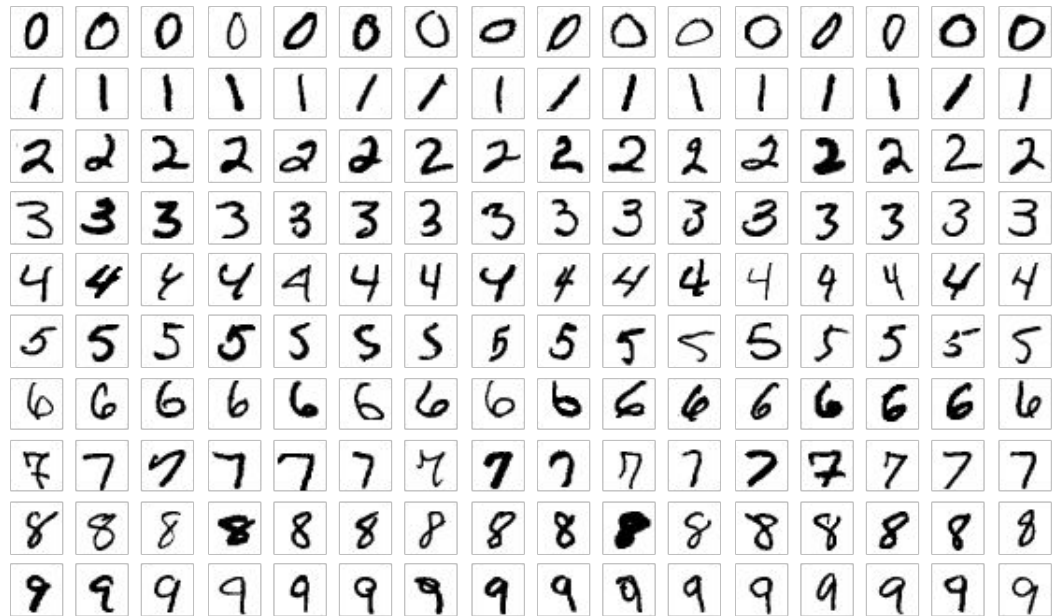
En mettant ensemble les informations pour chacun des patients, on obtient une grande matrice contenant toutes les informations :

16	67				
160	170				
59.0	73.9				
0	1				
110	122				
4.6	7.0				

Données

Si les données sont de plus grandes dimensions, comme des images par exemple, alors on a ce qu'on appelle des tenseurs : des matrices stackées les unes sur les autres.

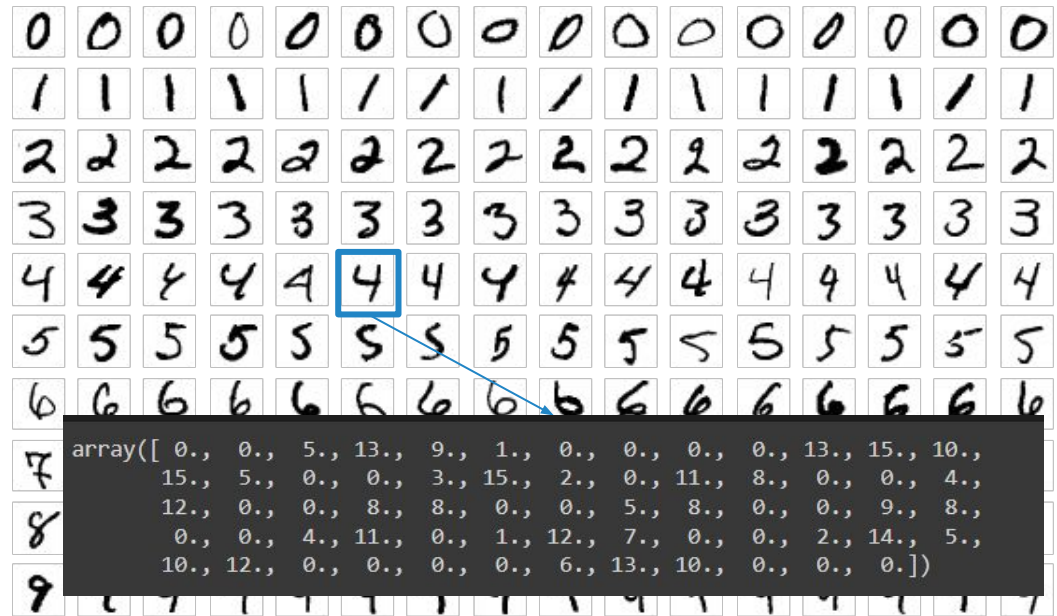
Exemple : MNIST



Données

Si les données sont de plus grandes dimensions, comme des images par exemple, alors on a ce qu'on appelle des tenseurs : des matrices stackées les unes sur les autres.

Exemple : MNIST



Implémentation en Python

On va utiliser deux bibliothèques différentes pour gérer nos données :

Numpy :

- Pour les vecteurs et les matrices
- `import numpy as np`
- Plein de fonctions sur les vecteurs et les matrices déjà implémentées : `np.add`, `np.dot` ...
- <https://numpy.org/doc/stable/>

Pandas :

- Pour gérer les données plus organisées
- `import pandas as pd`
- `df = pd.read_csv("....")`
- `Df['age']`
- Fonctions de gestion de dataset

Matrice - Implémentation

En informatique, matrice = Numpy array composé de numpy array.

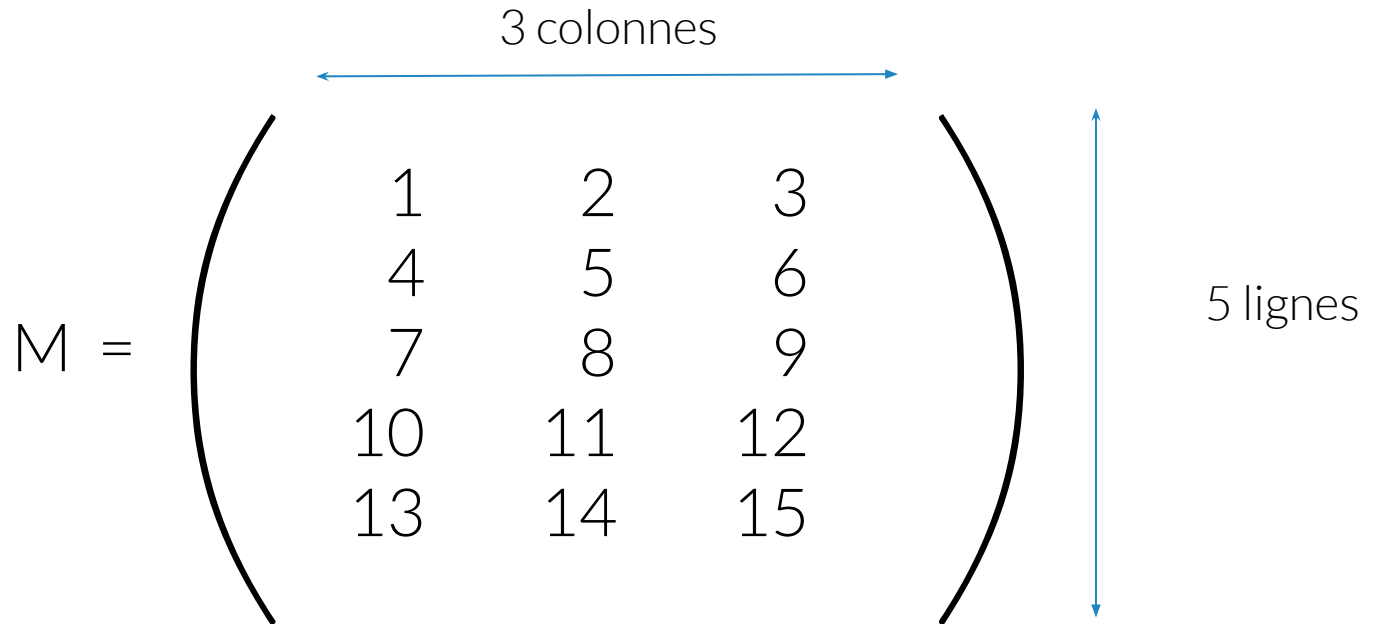
$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{pmatrix}$$

Matrice - Implémentation

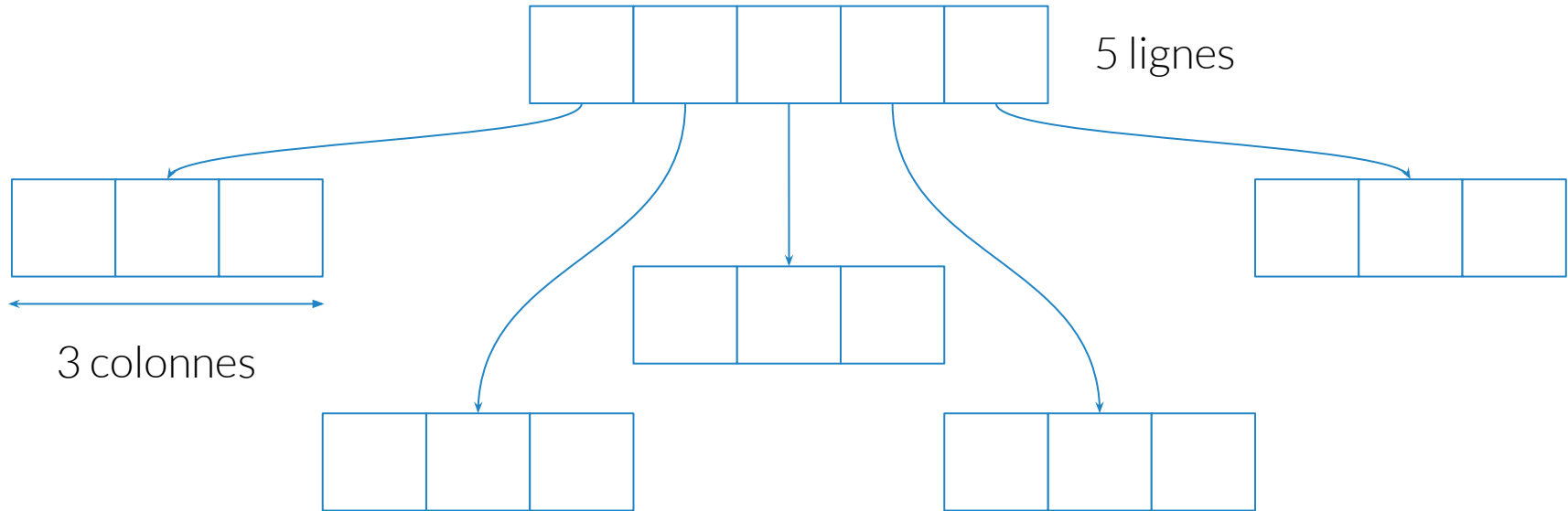
$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{pmatrix}$$

3 colonnes

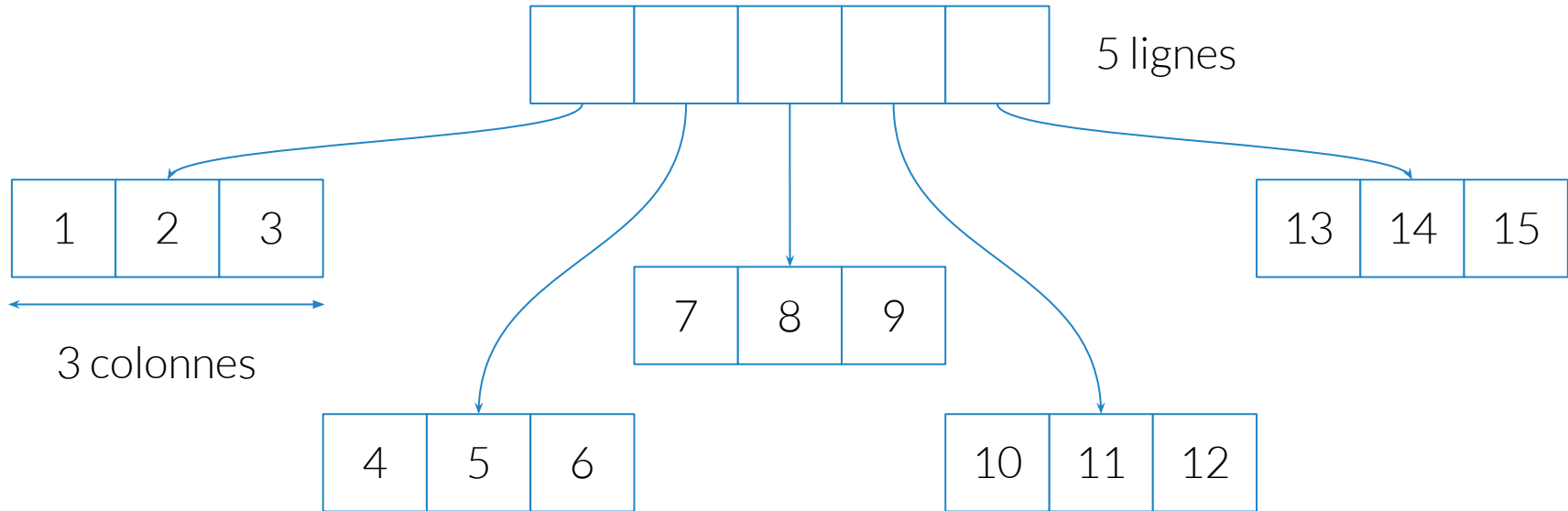
5 lignes

The diagram shows a matrix M with 5 rows and 3 columns. The elements are arranged in a 5x3 grid. A horizontal blue double-headed arrow above the matrix spans the width of the three columns and is labeled "3 colonnes". A vertical blue double-headed arrow to the right of the matrix spans the height of the five rows and is labeled "5 lignes".

Matrice - Implémentation



Matrice - Implémentation



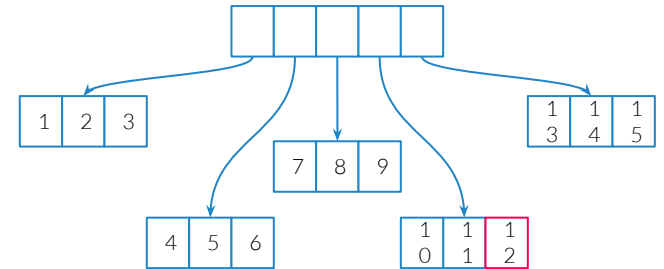
Matrice - Implémentation

M = ???

nombre de colonnes ?

nombre de lignes ?

accès au 3 éléments de la 4 eme ligne



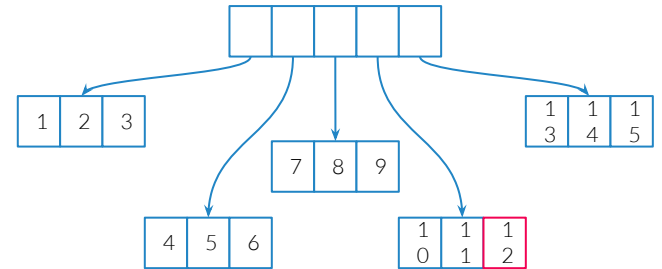
Matrice - Implémentation

M = ???

toute la première ligne

toute la deuxième colonne

première diagonale



Dataset - Données structurées

	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

âge	16
taille	160
poids	59.0
genre	0
BP	110
glycémie	4.6

Patient 1

âge	67
taille	170
poids	73.9
genre	1
BP	122
glycémie	7.0

Patient 2

Dataset - Données structurées

Dictionnaire :

Collection de données comme les listes, mais on accède aux éléments par des clés qui peuvent être autre chose que des indices. Associée à ses clés sont des valeurs.

Ici les clés sont 'data', 'target', 'frame', et 'feature_names'

```
{'data': array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
               [ 0.,  0.,  0., ..., 10.,  0.,  0.],
               [ 0.,  0.,  0., ..., 16.,  9.,  0.],
               ...,
               [ 0.,  0.,  1., ...,  6.,  0.,  0.],
               [ 0.,  0.,  2., ..., 12.,  0.,  0.],
               [ 0.,  0., 10., ..., 12.,  1.,  0.])),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
                  'pixel_0_1',
                  'pixel_0_2',
                  'pixel_0_3',
                  'pixel_0_4',
                  'pixel_0_5',
                  'pixel_0_6',
                  'pixel_0_7',
                  'pixel_1_0',
                  'pixel_1_1',
```


Dataset - Données structurées

```
D = {'data': array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
                  [ 0.,  0.,  0., ..., 10.,  0.,  0.],
                  [ 0.,  0., 10., ..., 12.,  1.,  0.])),
     'target': array([0, 1, 2, ..., 8, 9, 8]),
     'frame': None,
     'feature_names': ['pixel_0_0']}
}
```

```
D.keys() #acces à la liste des cles
```

```
D['data']
```

```
# Parcours du dictionnaire :
```

```
for cle in D :
```

```
    D[cle] #acces a la valeur correspondante a la cle
```

Dataset - Données structurées

Csv : comma-separated values

```
ID, Age, Sexe, Taille, Poids, BMI, BSA, HTA, DIAB, SBPo, DBPo, SBPh, DBPh, Glycemie
P0001, 16, 0, "160,0", "59,0", "23,0", "1,61", 0, 0, 110, 61, 106, "51,0", "4,6"
P0002, 18, 0, "157,0", "49,8", "20,2", "1,48", 1, 0, 112, 75, 135, "94,8", "4,6"
P0003, 19, 1, "173,0", "65,4", "21,9", "1,78", 1, 0, 144, 99, 154, "101,4", "4,7"
P0004, 22, 0, "159,5", "99,2", "39,0", "2,00", 1, 0, 160, 106, 172, "109,7", "5,1"
P0005, 22, 0, "171,0", "48,8", "16,7", "1,56", 1, 0, 136, 81, 149, "103,2", "4,1"
```

```
import pandas as pd
```

```
Df = pd.read_csv([NOM DU FICHER .CSV])
```

Dataset - Données structurées

```
import pandas as pd

#df est une DataFrame (un objet de pandas)
df = pd.read_csv([NOM DU FICHER .CSV])

df['Age']
df['Taille'].mean()
                .max()
                .min()

df.columns #liste des noms des colonnes

df['Sexe'].unique() #liste des valeurs que prend la colonne
```

Dataset - Données structurées

On a donc des données, plus ou moins structurées.

Il arrive que l'on veuille prédire certaines données en fonction d'autres.

- On veut prédire la glycémie en fonction de l'âge de la pression sanguine et de la taille
- On veut prédire un nombre en fonction de l'image d'un nombre
- On veut prédire une chanson à partir de toutes celles qui ont été écoutées avant
- On veut prédire la valeur suivante d'une série temporelle (bourse par exemple)

Ce que l'on veut prédire s'appelle un label (ou étiquette). On en dira plus en cours 3.

Indicateurs et visualisations



Que faire avec ces données ?

On a vu comment les données peuvent être représentées en machine.

Que faire avec ces données ?

Calcul d'indicateurs + visualisation de données

Indicateur - Vecteur

Exemple : Vecteur de tous les âges : `csv ['Age']`

Quelle est la population (en âge pour l'exemple mais ça peut être d'autres choses) dans notre base de données ?

Moyenne:

Soit une série X telle que $X = \{x_1, x_2, \dots, x_n\}$

La moyenne de X est :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Moyenne en python

```
def mean
```


Mediane, ecart type, variance

Soit une série X telle que $X = \{x_1, x_2, \dots, x_n\}$

Médiane : La valeur qui sépare la moitié inférieure de la moitié supérieure d'un ensemble. Souvent intéressant de comparer avec la moyenne

Variance : Mesure la dispersion d'une série autour de la moyenne

$$V = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 .$$

Mediane, ecart type, variance

Ecart type : Plus utilisé que la variance. On aura besoin de la variance pour calculer les covariances en Cours 2.

$$\sigma = \sqrt{V} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2}$$

L'écart type est la racine carrée de la variance.

Interprétation : En moyenne, une valeur de la série est éloignée de l'écart type de la moyenne de la série.

Calcul

```
mean_age = csv['Age'].mean()
median_age = csv['Age'].median()
std_age = csv['Age'].std()

print(mean_age, median_age, std_age)
```

```
55.26190476190476 57.0 13.760210921601534
```

```
all_age = csv['Age'].to_numpy()
```

```
mean_age = np.mean(all_age)
median_age = np.median(all_age)
std_age = np.std(all_age)

print(mean_age, median_age, std_age)
```

```
55.26190476190476 57.0 13.745310851263282
```

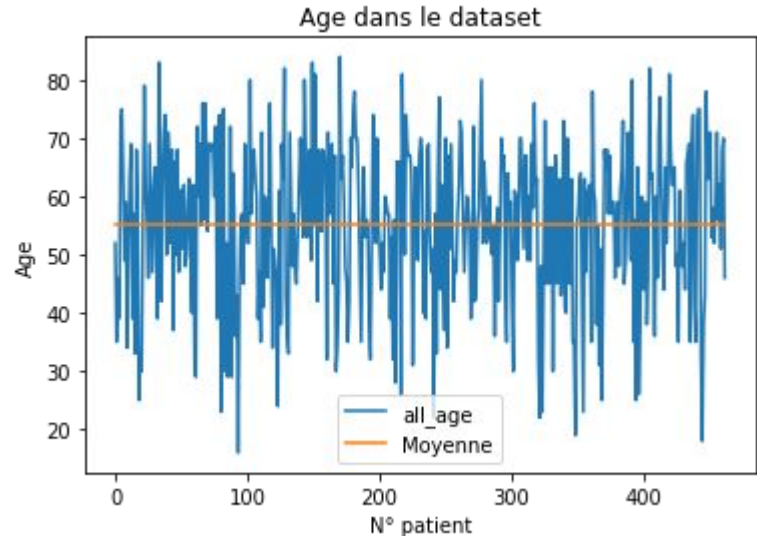
Visualisation

On utilise matplotlib pour créer des graphiques en python
(<https://matplotlib.org/>).

```
import matplotlib.pyplot as plt

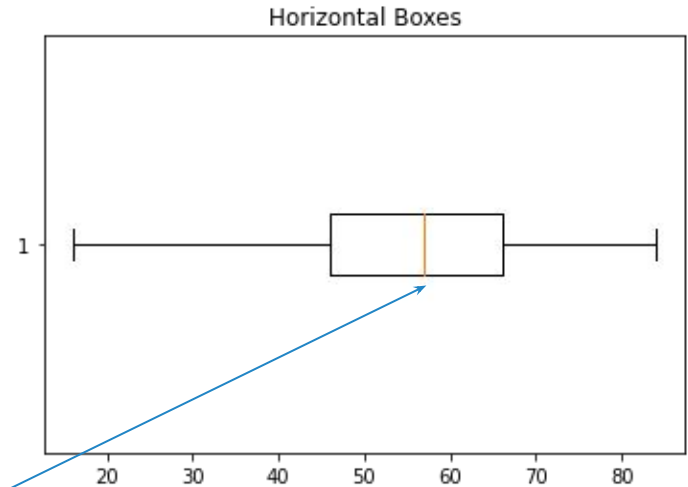
plt.plot(all_age, label="all_age")
plt.plot([mean_age]*len(all_age),
label="Moyenne")

plt.title('Age dans le dataset')
plt.xlabel('N° patient')
plt.ylabel('Age')
plt.legend()
plt.show()
```



Visualisation 2 - boite a moustache

```
fig, ax = plt.subplots()
ax.set_title('Horizontal Boxes')
ax.boxplot(all_age, vert=False)
```

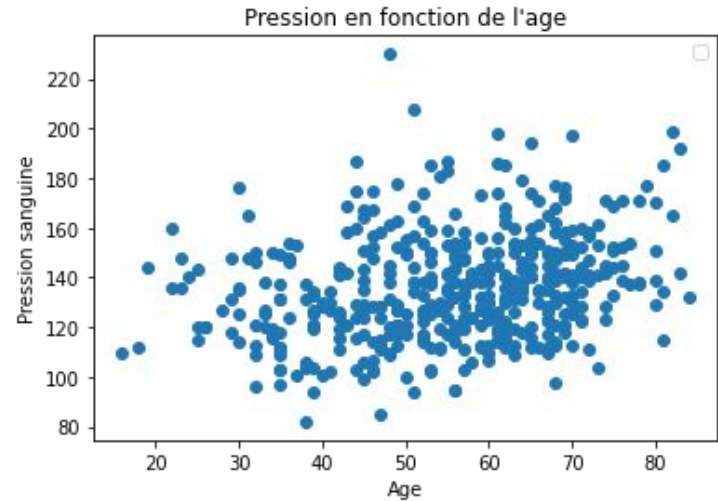


Moyenne

Visualisation 3

Visualisation d'une données en fonction d'une autre :

```
plt.scatter(csv['Age'], csv['SBPo'])  
plt.title('Pression en fonction de l\'age')  
plt.xlabel('Age')  
plt.ylabel('Pression sanguine')  
plt.legend()  
plt.show()
```



Visualisation 4

```
plt.scatter(csv['Age'], csv['SBPo'], c=csv['Sexe'])  
plt.title('Pression en fonction de l\'age')  
plt.xlabel('Age')  
plt.ylabel('Pression sanguine')  
plt.legend()  
plt.show()
```

