UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

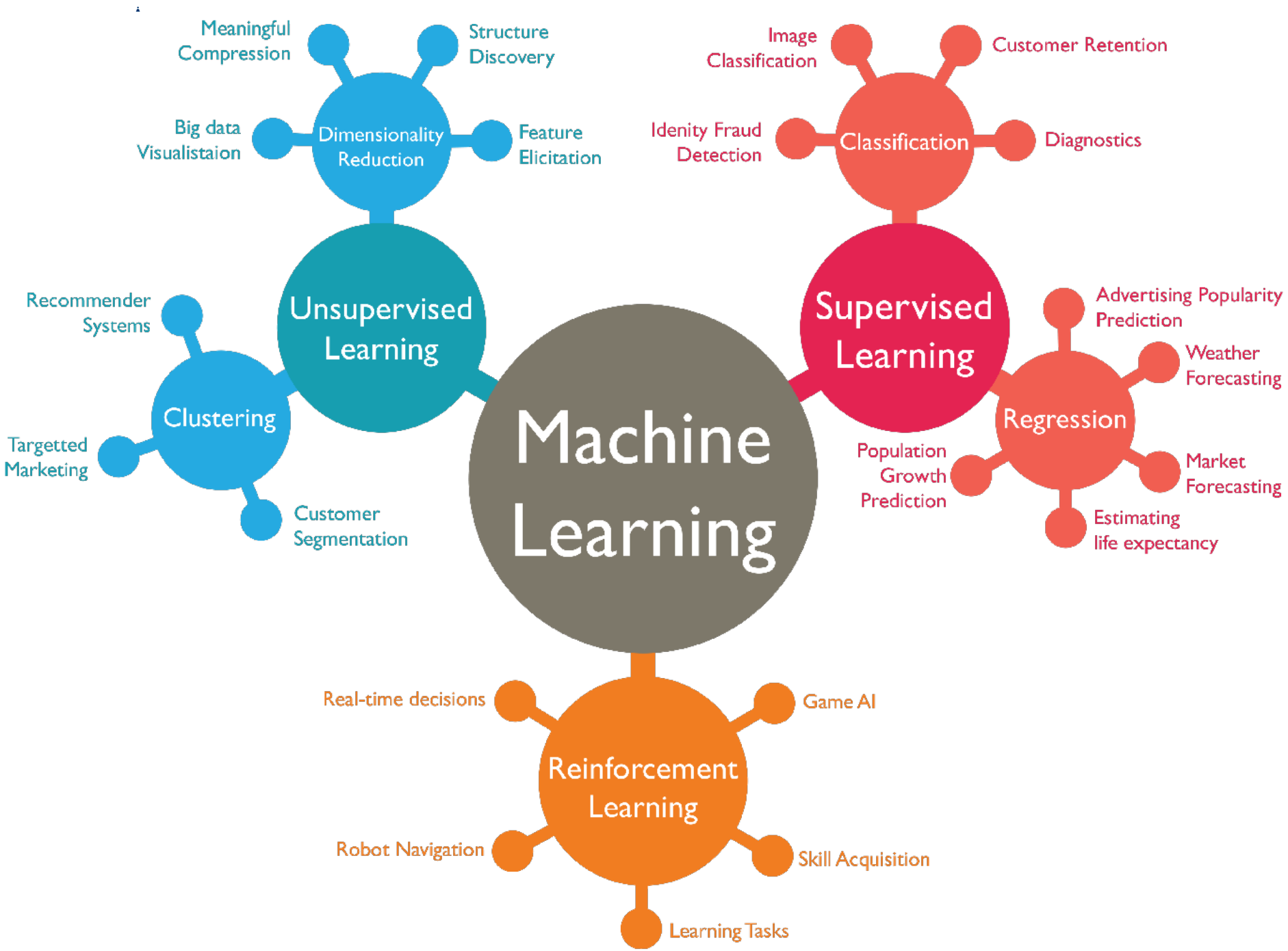
Clustering

(= Classification non-supervisée)

Fichiers sur

https://github.com/mkirschpin/CoursPython

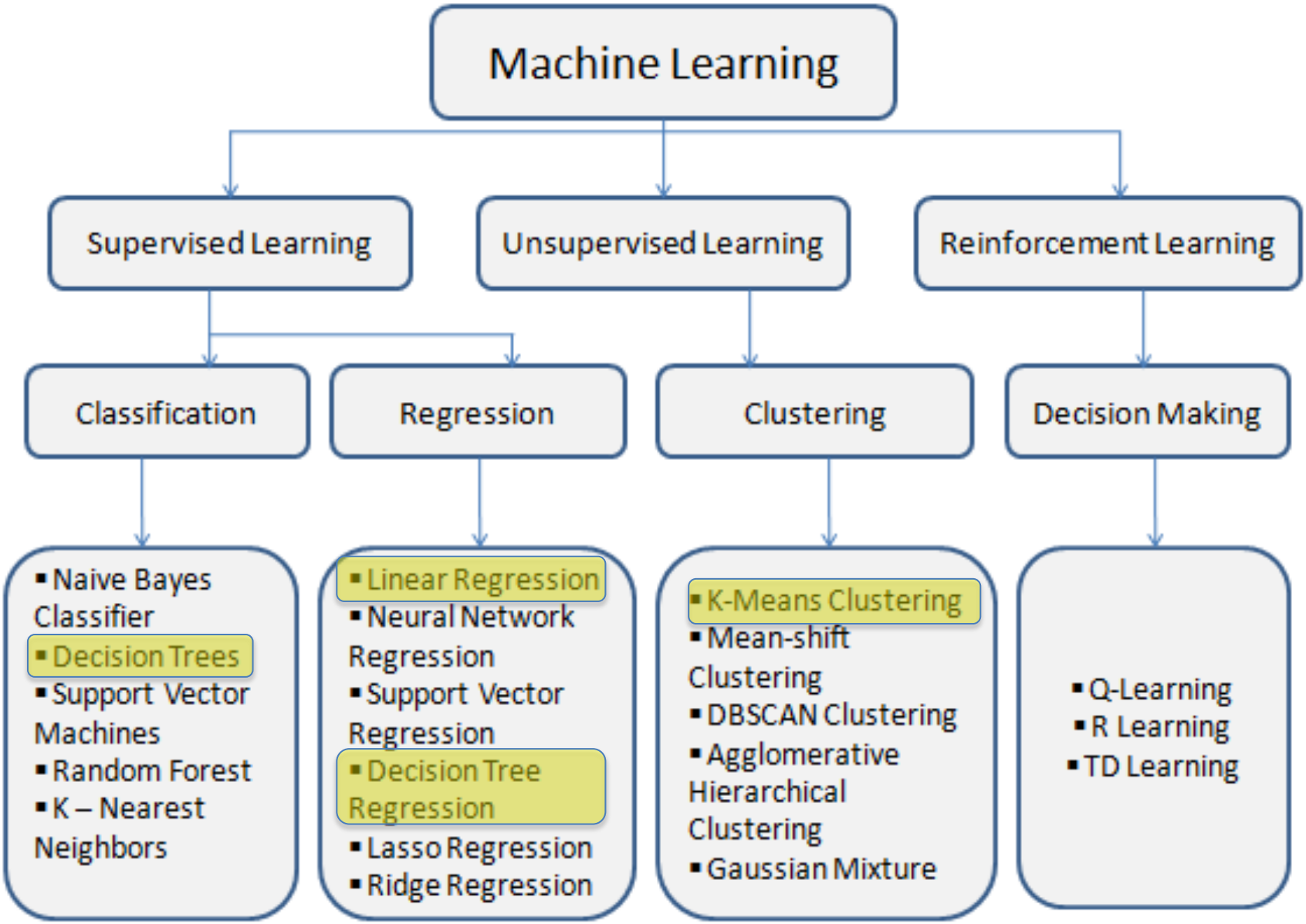
http://kirschpm.fr/cours/PythonDataScience/



https://wordstream-files-prod.s3.amazonaws.com/s3fs-public/machine-learning.png

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT

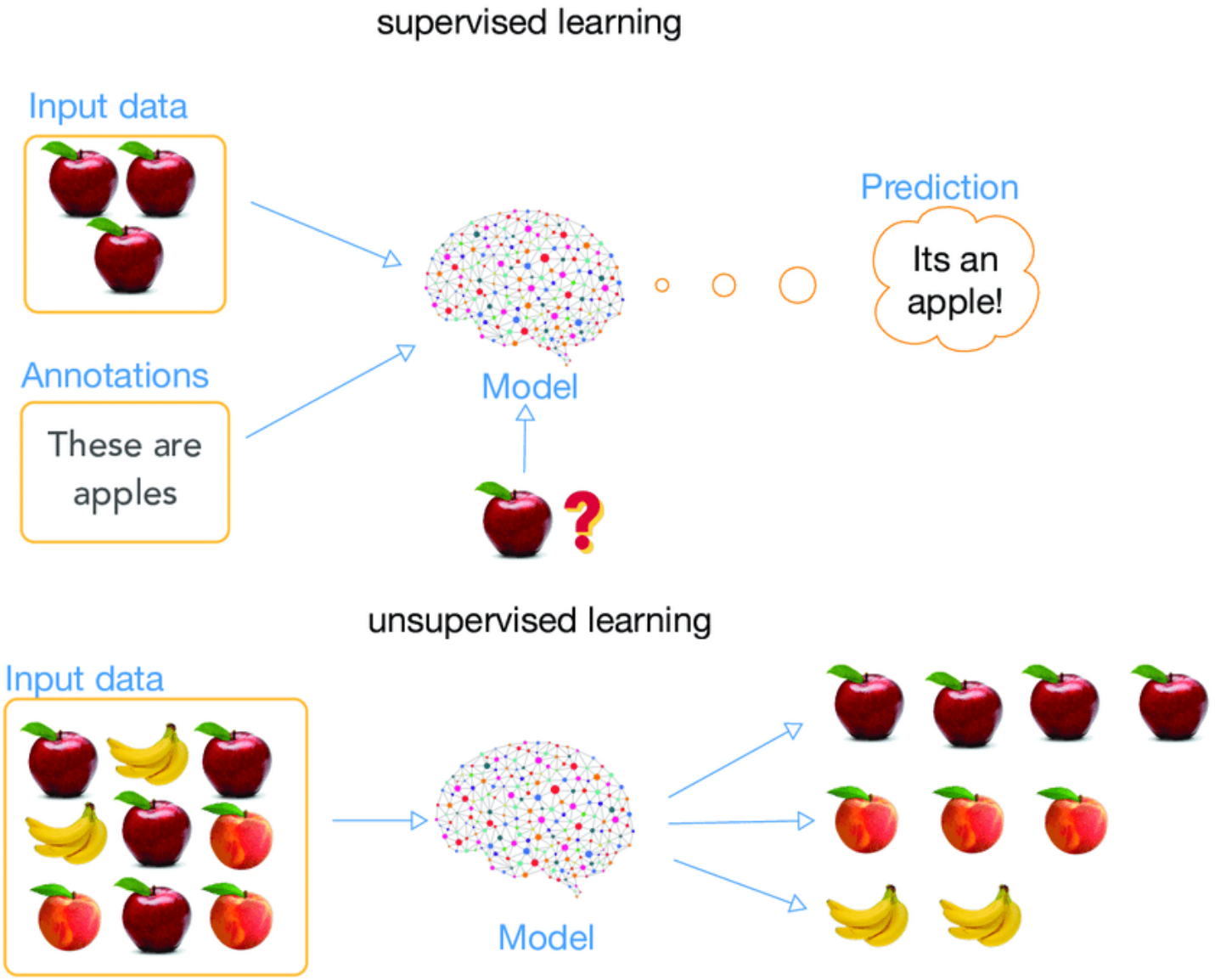
Types of Machine Learning



https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Apprentissage supervisé / non supervisé

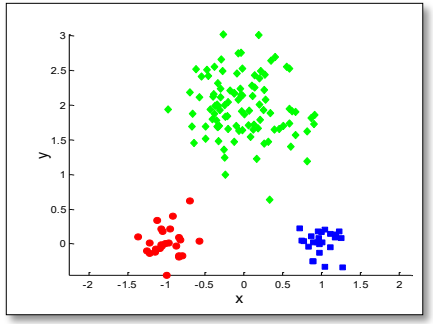


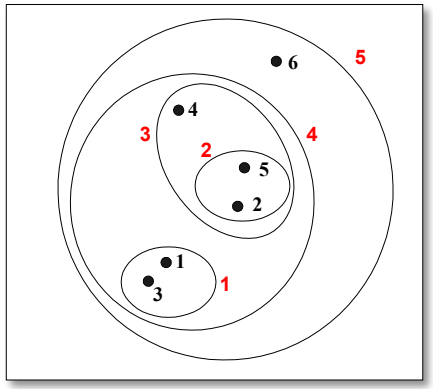
Ma, Y., Liu, K., Guan, Z., Xu, X., Qian, X., & Bao, H. (2018). Background augmentation generative adversarial etworks (BAGANs): Effective data generation based on GAN-augmented 3D synthesizing. *Symmetry*, *10*(12), 734.

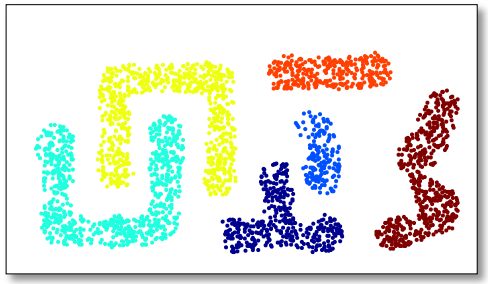
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Trois grands types de clustering

- **Clustering par partition**
  - Une division des données en **sous-ensembles** (***clusters***) **disjoints**
- **Clustering hiérarchique**
  - Un ensemble de ***clusters*** **emboités** les uns dans les autres, avec une **structure hiérarchique**
- **Clustering par densité**
  - Une division des données en ***clusters*** **disjoints** qui s'appuie sur la **densité** estimée des clusters





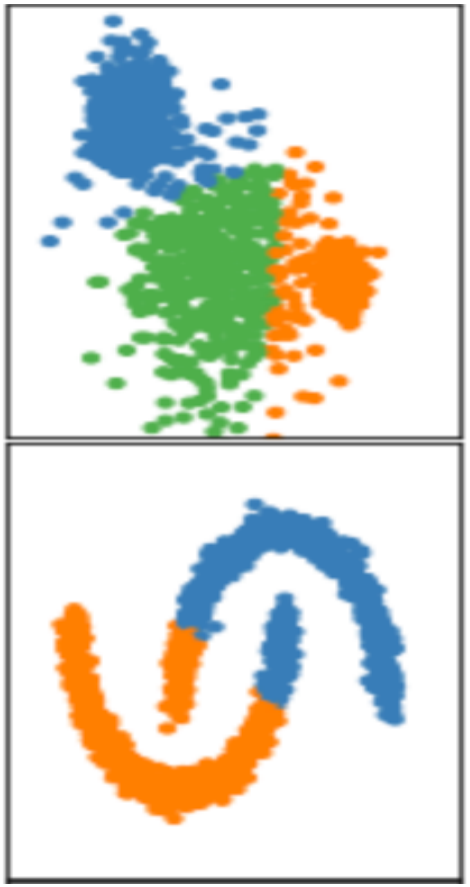


UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Clustering par Partition (le plus utilisé)

- Une bonne méthode de **regroupement**
- Permet de garantir :
  - Une **grande** **similarité** **intra-groupe**
  - Une **faible** **similarité** **inter-groupe**

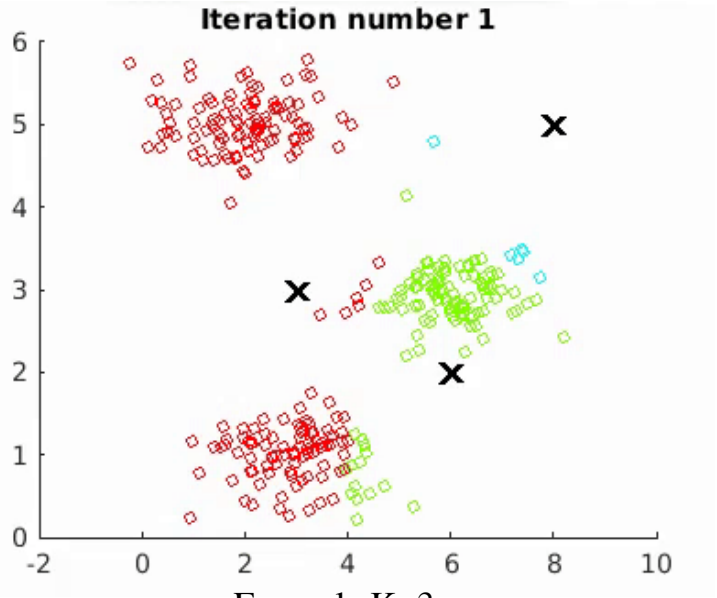
- **Quelques limitations :**
  - N'est pas adapté à des "formes"complexes
  - Peut produire des résultats différents à chaque exécution



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Fonctionnement K-means

- Prendre **K** points (au hasard) comme centroïdes initiaux
- Répéter
  - Former K clusters en assignant les points à leur centroïde le **plus proche**
  - Mettre à jour le centroïde de chaque cluster jusqu'à ce que qu'aucun centroïde ne bouge



Exemple K=3

- **Notion de distance**
  - Chaque cluster contient **les points les plus proches**
    - avec la **métrique MSE**
  - **Aucune étiquette n'est nécessaire**

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

K-means

```
from sklearn.cluster import KMeans
```

Création objet modèle

```
means = KMeans(n_clusters=3)
```

*On doit indiquer le nombre de clusters **K***

*On **construit** le modèle (**fit**)*

```
means.fit(dataframe)
```

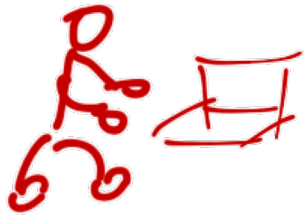
Dataframe donné en entrée
(sans colonne 'target' car apprentissage non supervisé)

```
means.labels_
```

*On **affiche** les labels (clusters) attribués à chaque valeur*

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands On !



- **Etude du dataset**
  - Créer le **DataFrame avec les données Iris**

```
from sklearn.datasets import load_iris
import pandas as pnd


iris = load_iris()
iris_df = pnd.DataFrame(iris.data, columns=iris.feature_names)


iris_df['target'] = iris.target


iris_df.head()
```

On n'oublie pas les **imports**

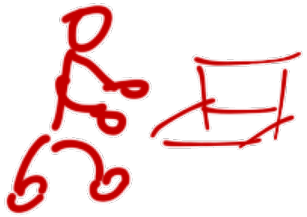
Création du DataFrame

Cette colonne indiquant l'espèce ne sera utilisée que pour comparer les résultats aux labels après coup

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

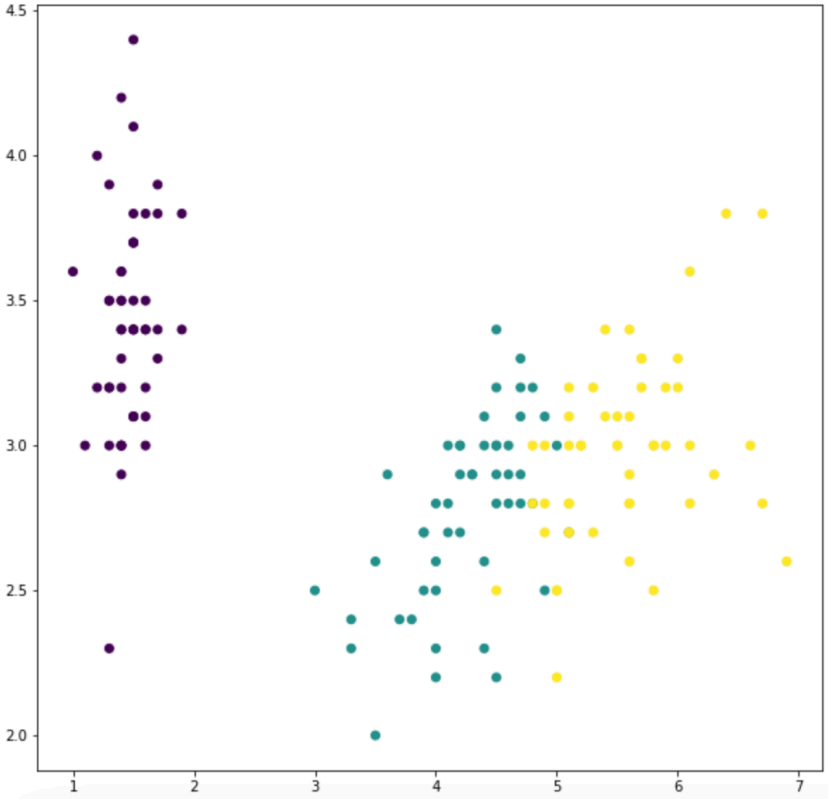
Hands On !



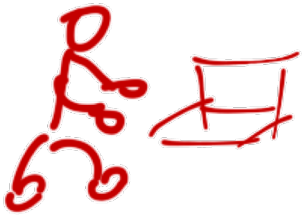
- Visualisation des données
  - On peut afficher les données sur un plan 2D
    - Problème : on a 12 combinaisons possibles (4 *features* x 3 classes d'Iris)

```
import matplotlib.pyplot as plt
%matplotlib inline


plt.figure(figsize=[10,10])

plt.scatter ( iris_df[‘petal length (cm)’],
              iris_df['sepal width (cm)']
              c=iris_df['target'])
```

Création d'une figure taille 10x10

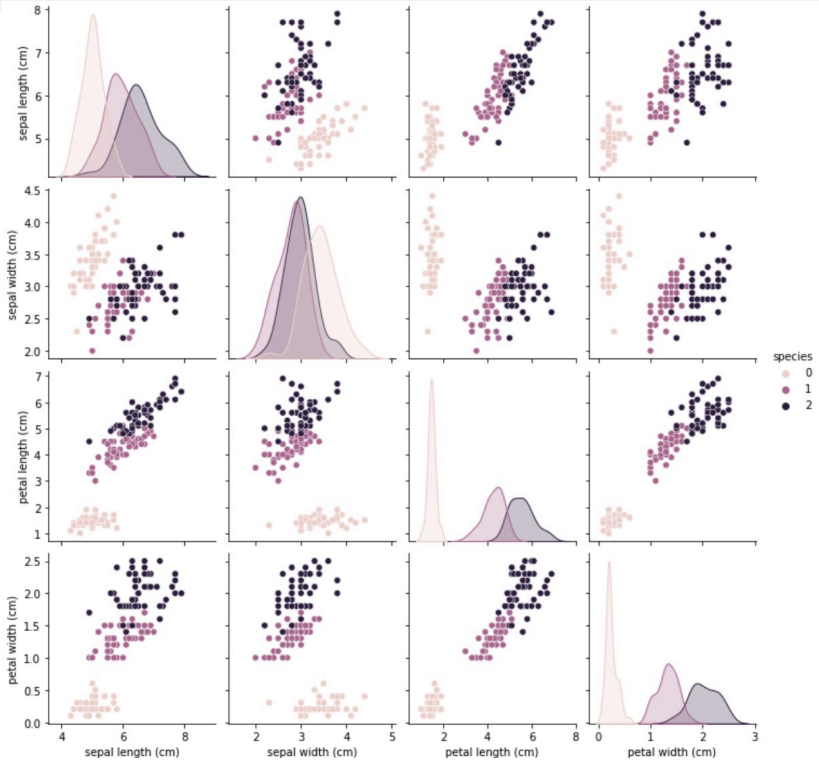


Hands on !



- Visualisation des paires de features avec Seaborn
  - Affichage des combinaisons des différentes *features* 2 à 2

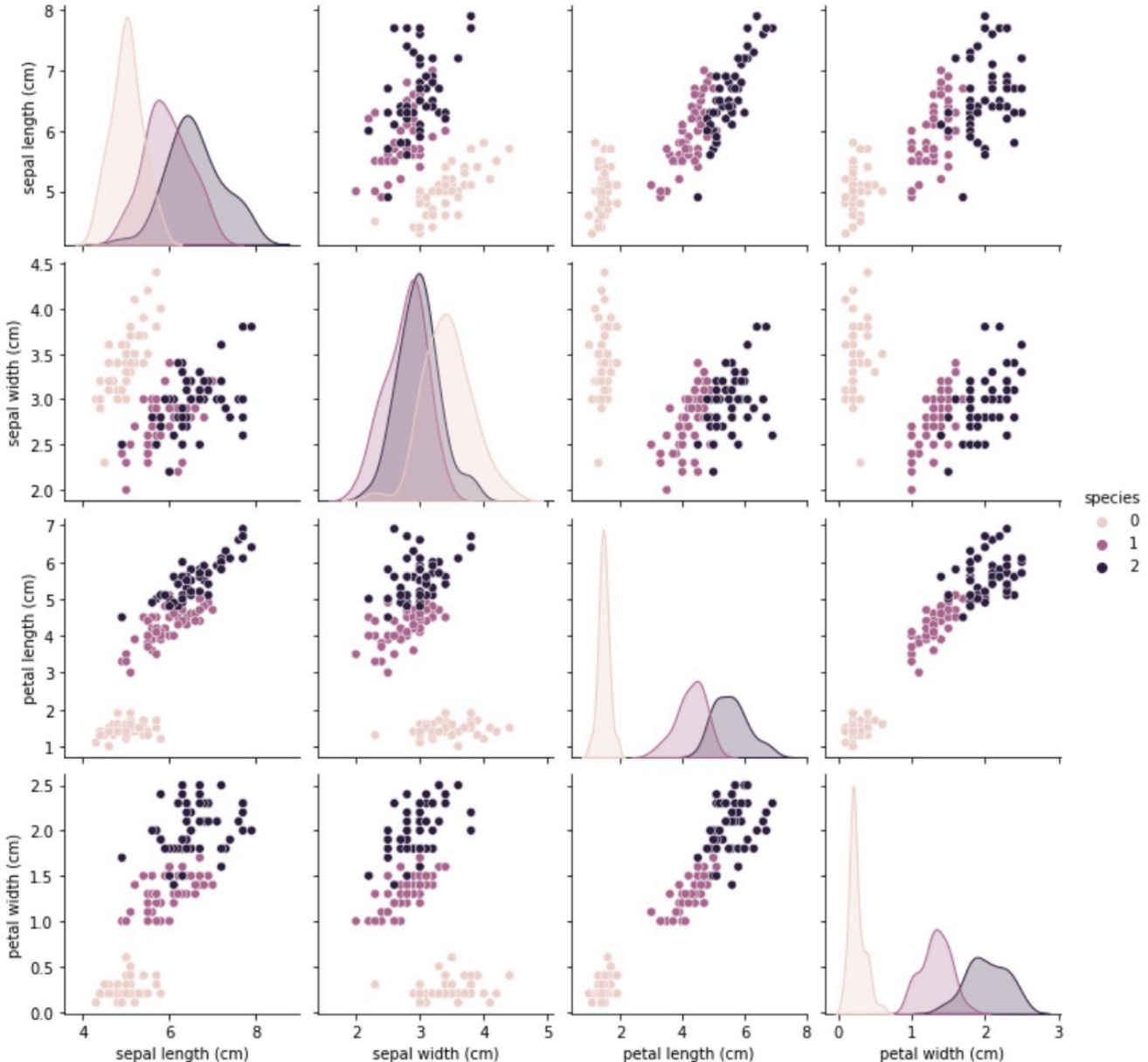
```
import seaborn as sns
grid = sns.pairplot(data=iris_df, vars=iris_df.columns[0:4], hue='target')
```



  - Ici on peut observer que certaines combinaisons de *features* sont plus propices à la séparation en clusters

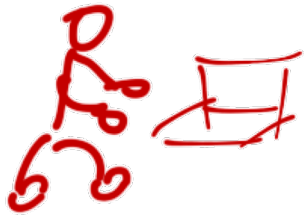
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands on !



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands On !



- **Exemple clustering**
  - Créer le modèle K-Means
  - Entraîner le modèle

On fait d'abord l'import

```
from sklearn.cluster import KMeans
```

Création du modèle
avec 3 classes (K = 3)

```
kmeans = KMeans ( n_clusters=3 , random_state=11 )
```

***Random_state=11** pour obtenir toujours le même résultat*

Entrainement du modèle

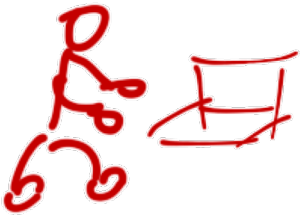
```
kmeans.fit( iris_df.drop(['target'], axis='columns')  )
```

***Important :** modèle non supervisé, donc on ne fournit pas de target*

On supprime la colonne target

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands On !



- **Exemple clustering**
  - Afficher les labels créés lors du clustering :

```
print(kmeans.labels_)
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```

On constate que certains 0 et certains 2 sont mélangés

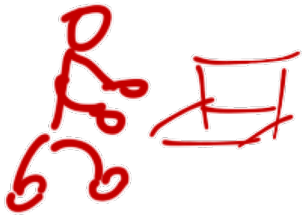
  - Comparaison avec les target « officiels » :

```
print(iris.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands On !



- **Suite de l'Exercice : faire un clustering des espèces d'Iris**
  - On peut afficher les points colorés selon les **labels estimés**

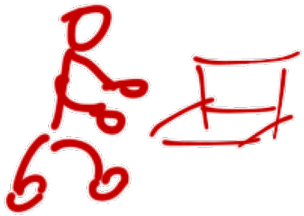
```
plt.scatter(iris_df['petal length (cm)'], iris_df['sepal width (cm)'],
            c=kmeans.labels_ )

plt.show()
```

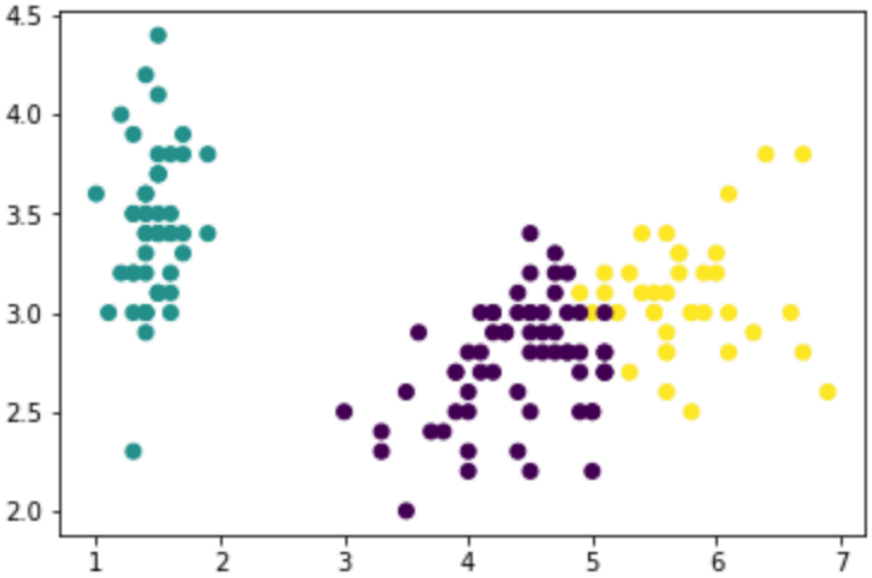
  - Faisons le même, cette fois-ci montrant les **labels d'origine** (*iris.target*)

```
plt.scatter(iris_df['petal length (cm)'], iris_df['sepal width (cm)'],
            c=iris_df['target'] )

plt.show()
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

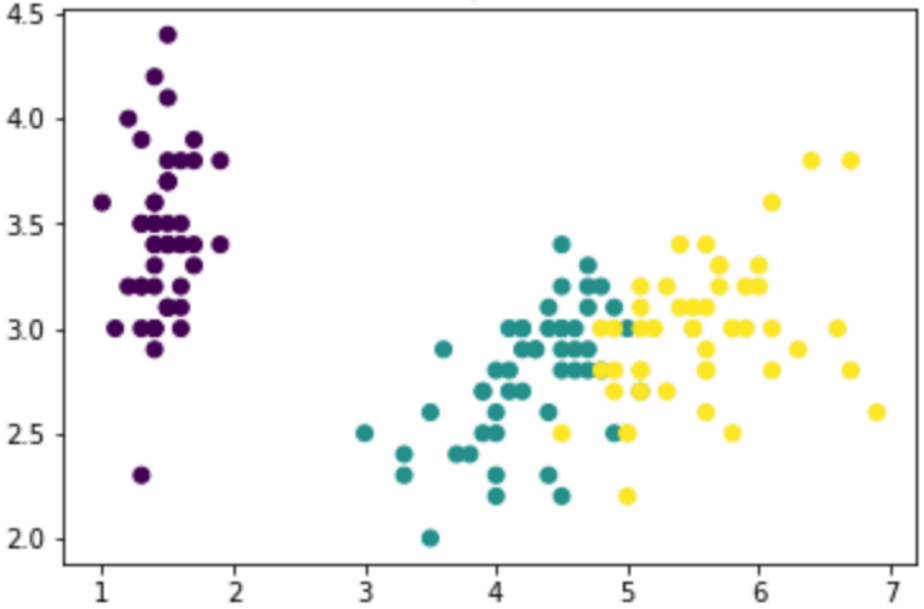
Hands On !





Labels estimés

Labels d'origine



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

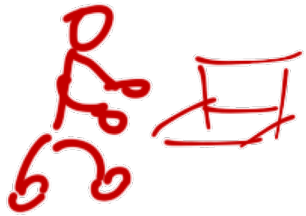
Comment choisir le nombre de clusters?

- Parfois, on ne connait pas la « bonne » valeur de K
  - On peut déterminer de manière expérimentale en observant :
    - le SSE (somme des erreurs au carré) – méthode "coude" (elbow)
    - ou le score "silhouette" – mesure la cohésion des clusters

```
from sklearn import metrics
import numpy as np

#utilisation de la métrique "silhouette"
#faire varier le nombre de clusters de 2 à 10
res = np.arange(9,dtype="double")
for k in np.arange(9):
    km = KMeans(n_clusters=k+2)
    km.fit( iris_df.drop(['target'], axis='columns') )
    res[k] = metrics.silhouette_score(iris_df.drop(['target'],axis='columns'),km.labels_)
print(res)

import matplotlib.pyplot as plt
plt.title("Silhouette")
plt.xlabel("# of clusters")
plt.plot(np.arange(2,11,1),res)
plt.show()
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Hands On !



Testez d'autres valeurs de K
Testez d'autres algorithmes de clustering
CAH

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

CAH
Classification Ascendante Hiérarchique

- Méthode agglomérative
- Voir notebook kmeans-silhouette-cah-fromage.ipynb

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Aide pour le projet

- Voir notebook creation-colonne-calculee.ipynb