Data Science Bootcamp

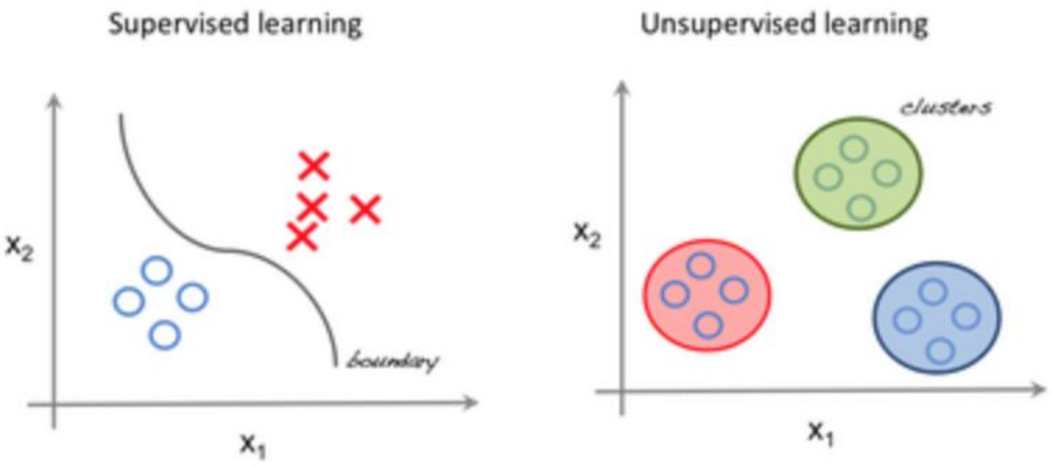
Machine Learning non supervisé

Jedha

Cas d'usage du ML non supervisé

- **Réduction de dimensions →** Compression de données
- **Clustering →** Diviser les données en groupes cohérents

Classification vs. Clustering



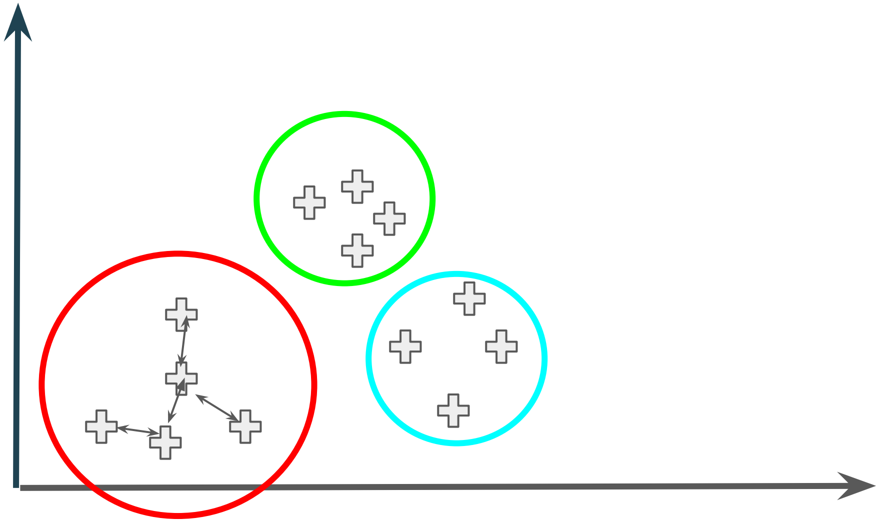
KMeans

L’algorithme KMeans

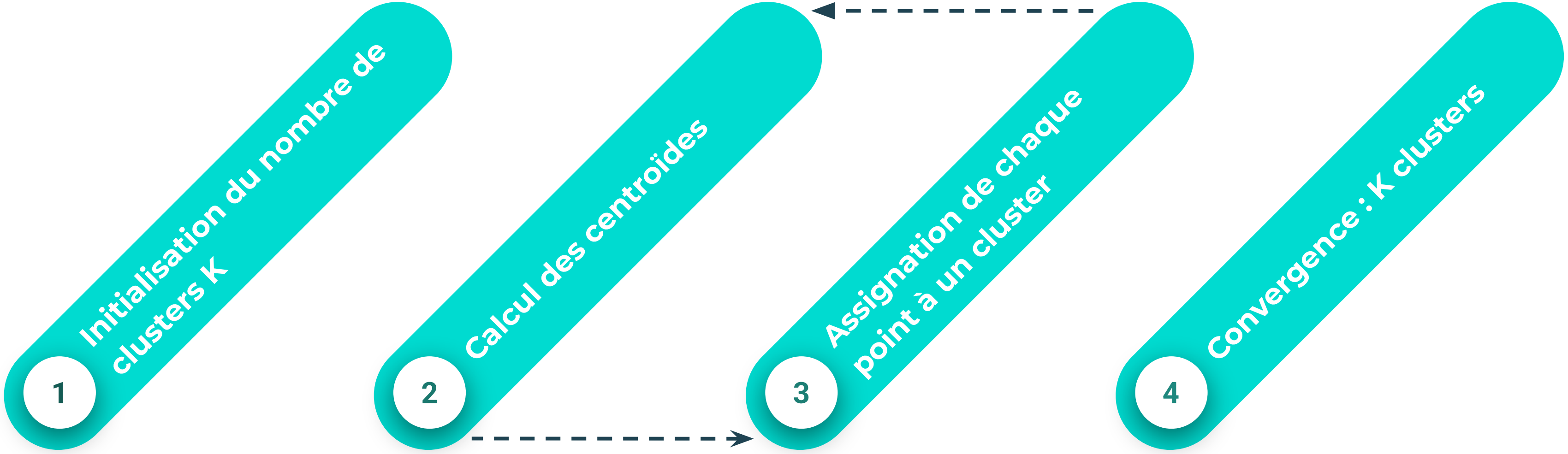
L’algorithme **KMeans** sépare les données en groupe de variance égale, en minimisant un critère appelé inertie (**inertia**) ou distance-intra-cluster (**wcss : within- cluster sum-of-squares**)

Source: Sklearn

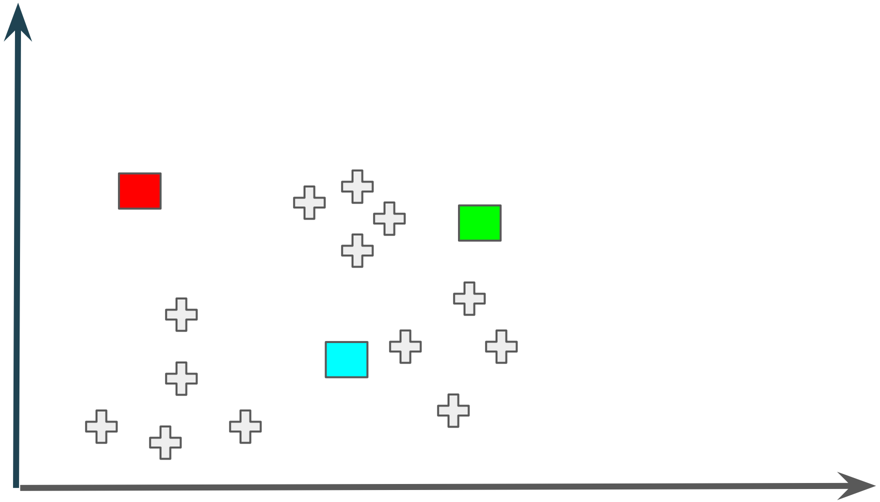
KMeans Algorithm



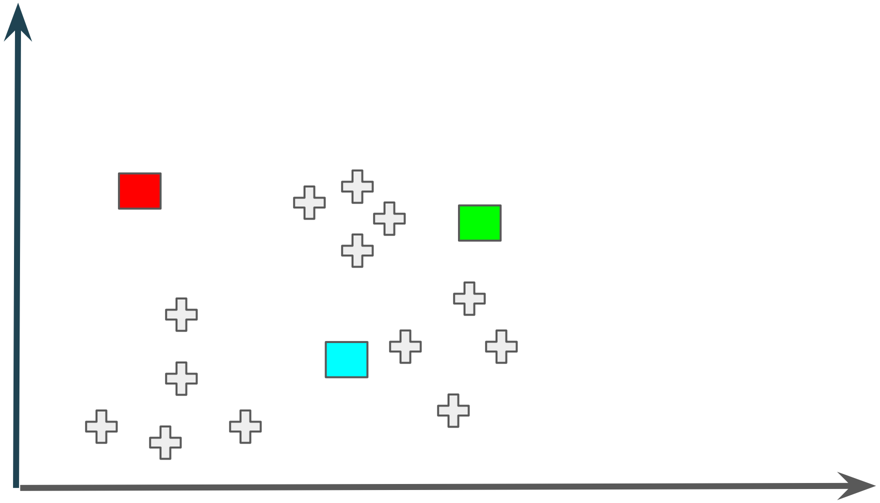
Principe


Initialisation du nombre de clusters K
1
Calcul des centroïdes
2
Assignation de chaque point à un cluster
3
Convergence : K clusters
4

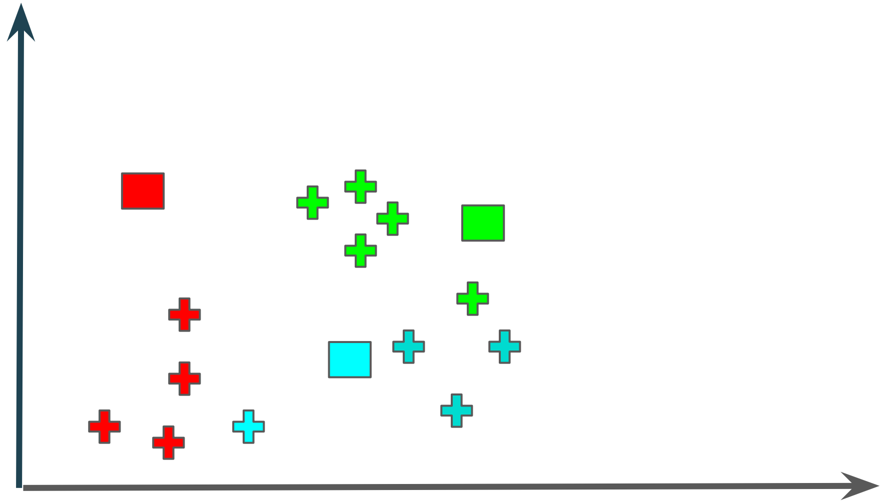
Etape 1 – Initialisation de K clusters



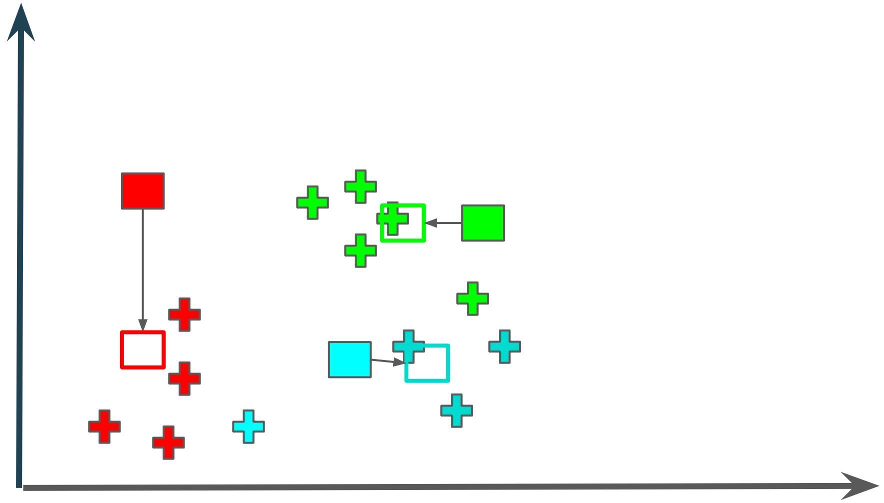
Etape 2 - Calcul des centroïdes



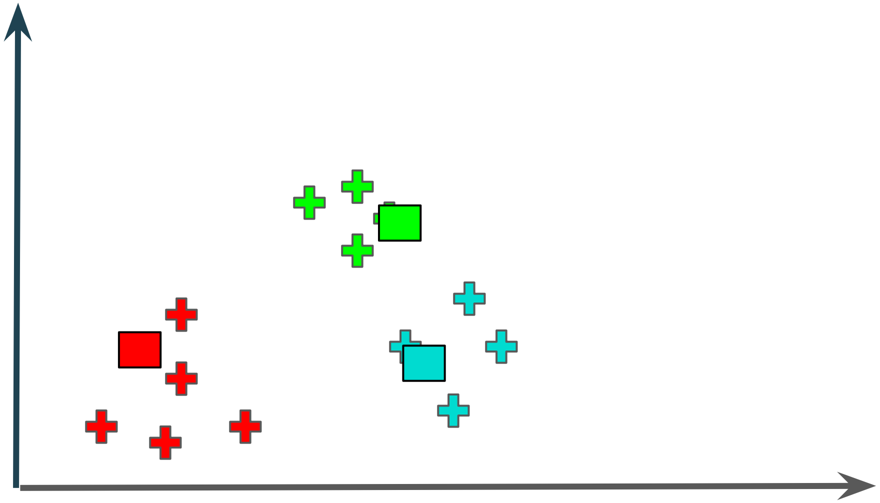
Etape 3 – Assignation des points au centroïde le plus proche



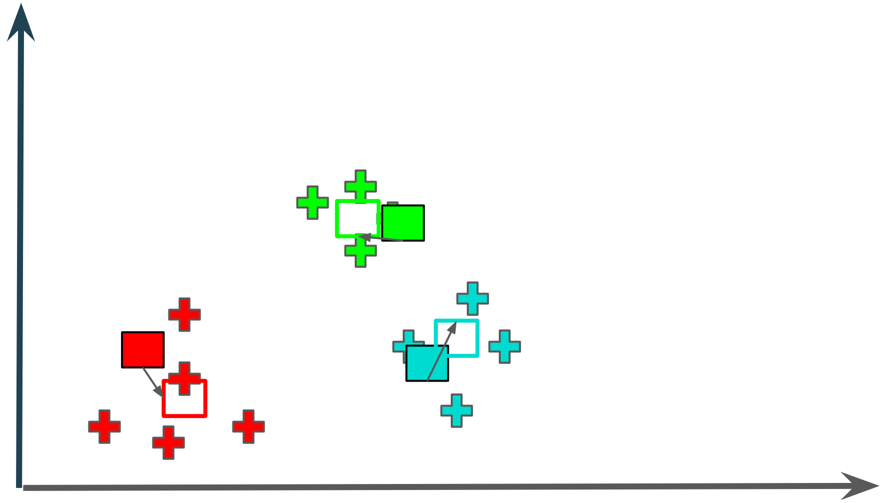
Etape 2 – Calcul des centroïdes



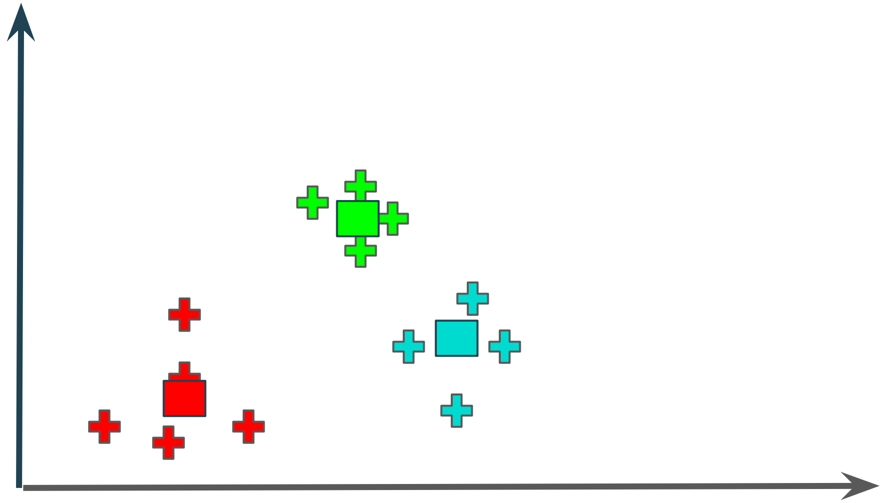
Etape 3 – Assignation des points au centroïde le plus proche



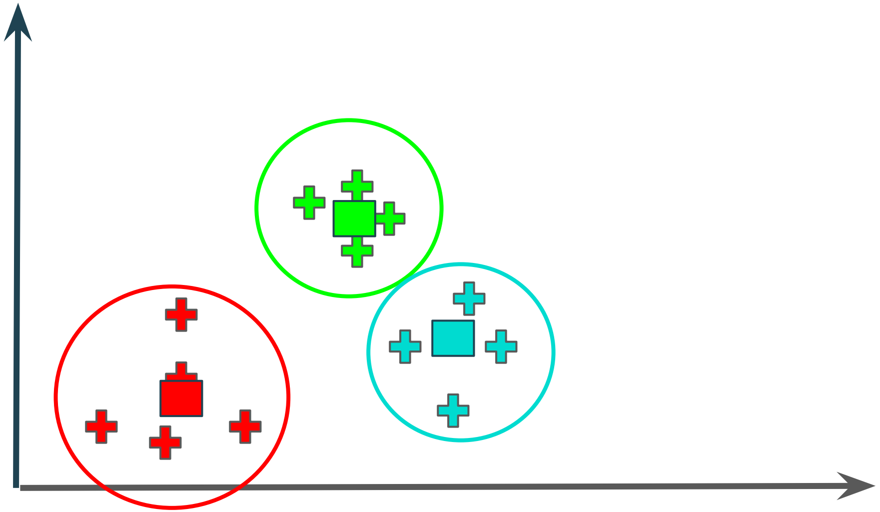
Etape 2 – Calcul des centroïdes



Etape 3 – Assignation des points au centroïde le plus proche



Etape 4 – Convergence : les clusters ne changeront plus

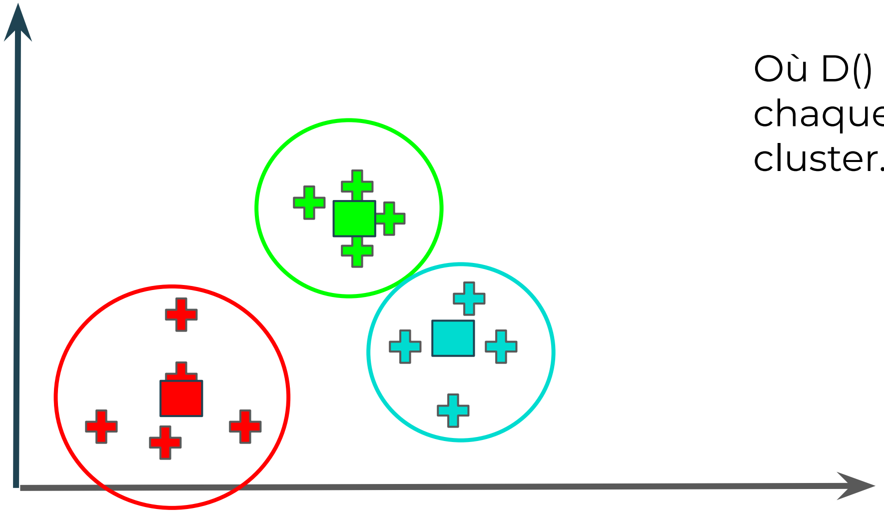


Comment choisir K, le nombre de clusters ?

Deux méthodes complémentaires

- **Elbow** → See if data points within a cluster are close from the centroid
- **Silhouette** → See if clusters are far from each other

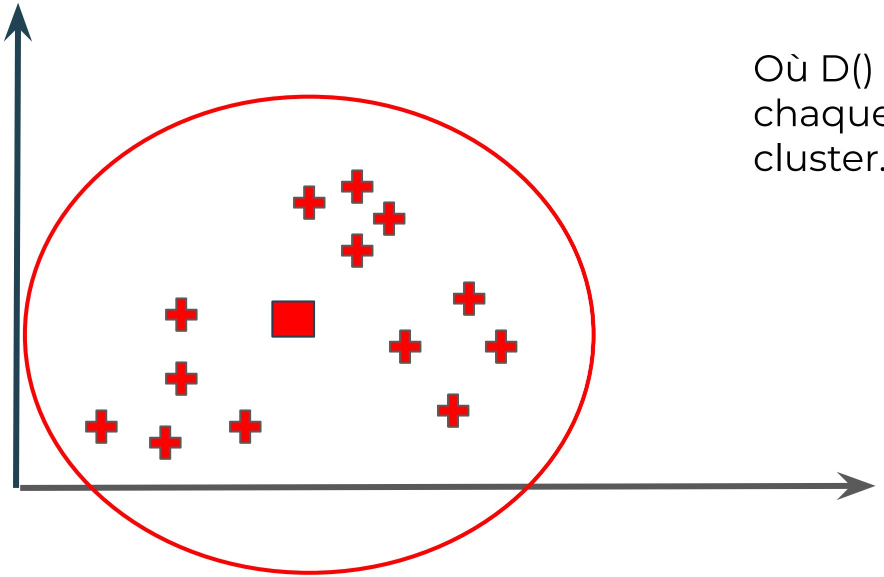
Elbow



WCSS = D(✚, ■) + D(✚ , ■) + D(✚, ■)

Où D() est la distance euclidienne entre chaque point et le centroïde de son cluster.

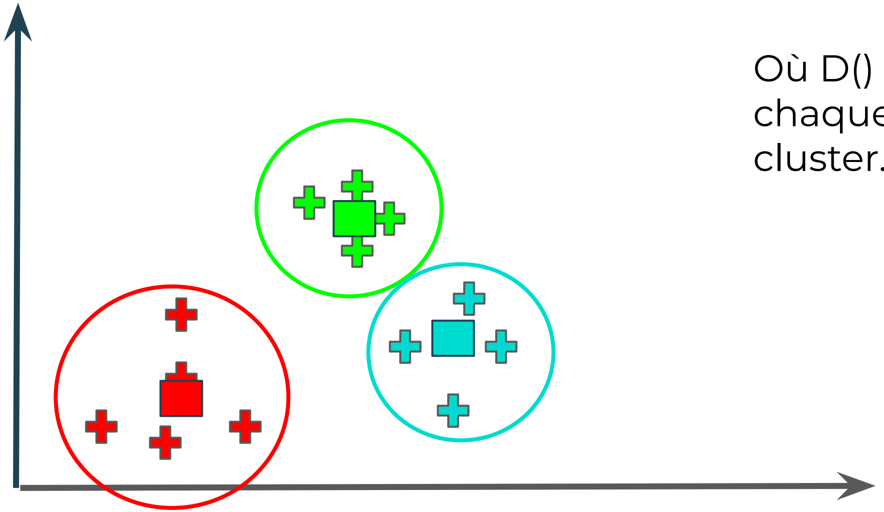
Elbow - WCSS pour K=1



WCSS = D(, )

Où D() est la distance euclidienne entre chaque point et le centroïde de son cluster.

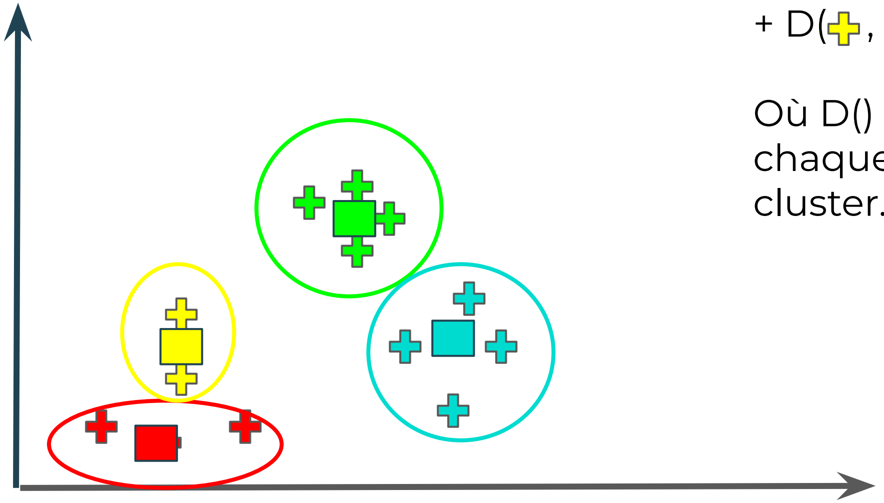
Elbow - WCSS pour K=3



WCSS = D(✚, ■) + D(✚ , ■) + D(✚, ■)

Où D() est la distance euclidienne entre chaque point et le centroïde de son cluster.

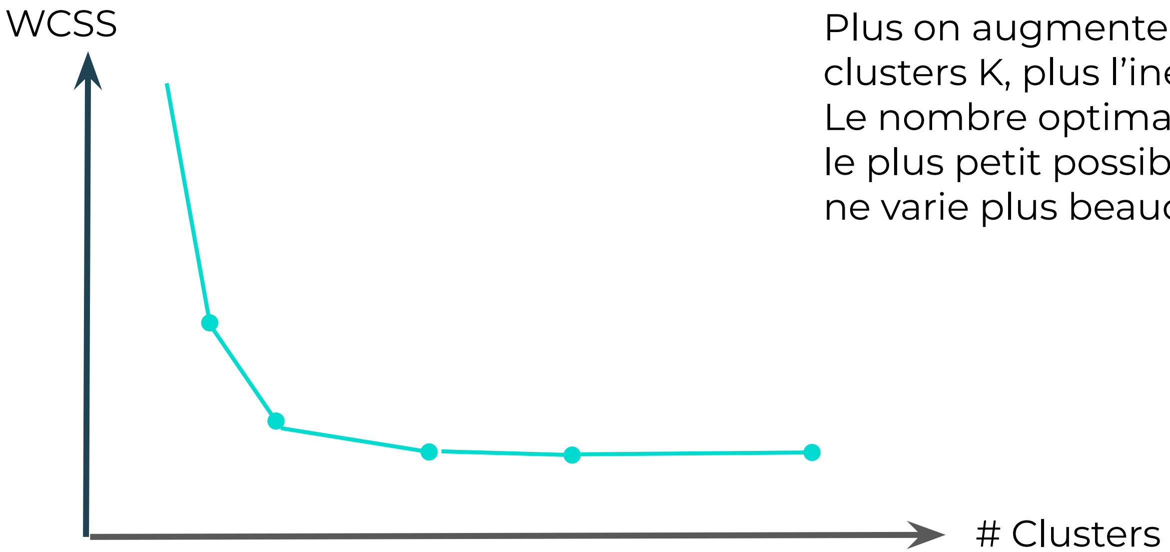
Elbow - WCSS pour K=4



WCSS = D(,) + D(,) + D(,)
+ D(,)

Où D() est la distance euclidienne entre chaque point et le centroïde de son cluster.

Elbow – Tracer WCSS en fonction de K



Plus on augmente le nombre de clusters K, plus l'inertie (WCSS) diminue. Le nombre optimal de clusters est le K le plus petit possible pour lequel l'inertie ne varie plus beaucoup.

Silhouette

$$\frac{b^i - a^i}{max(a^i, b^i)}$$

Où

a → distance moyenne entre le point i et tous les points du même cluster
b → distance moyenne entre le point i et tous les points du cluster le plus proche

Silhouette

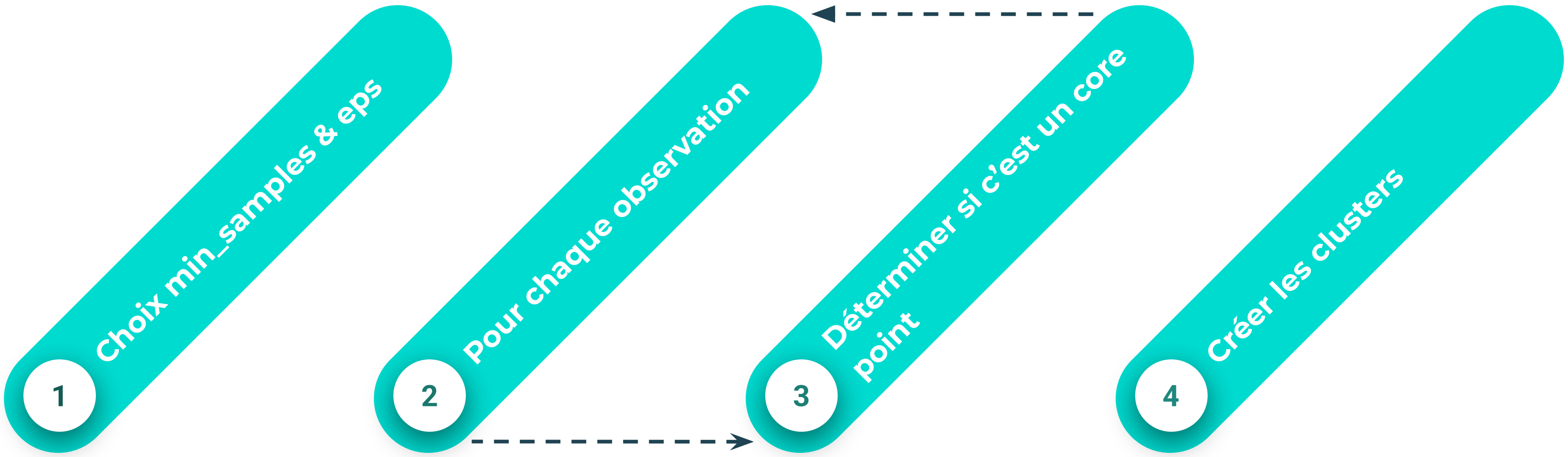
- **Proche de 0** → Clusters très proches les uns des autres
- **Proche de 1** → Clusters très éloignés les uns des autres

DBSCAN

Une approche différente

- Algorithme basé sur la notion de densité
- DBSCAN crée des clusters correspondant à des zones à forte densité de points
- Pas besoin de choisir le nombre de clusters K : DBSCAN le fait automatiquement
- Gestion des outliers : les points isolés sont rejetés

Principe

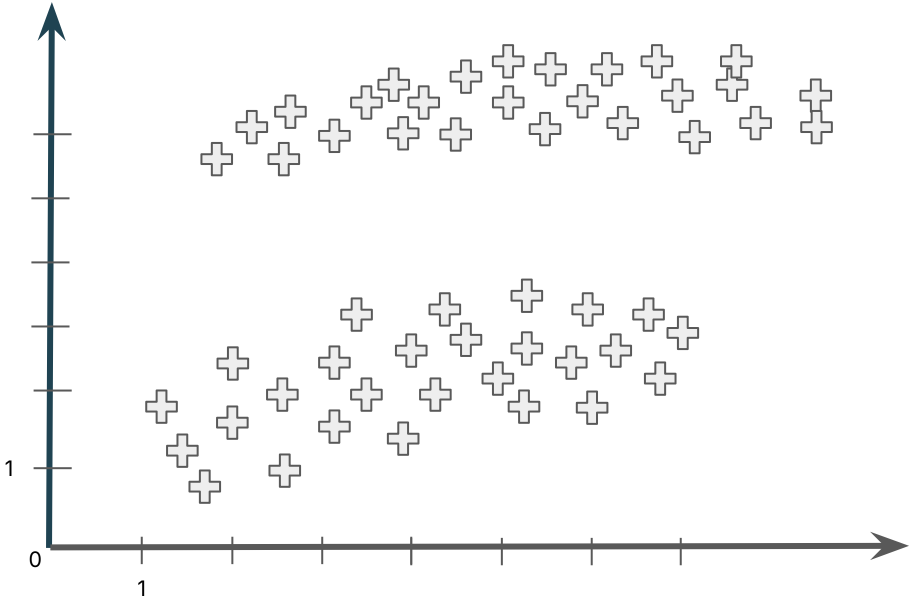

1
Choix min_samples & eps
2
Pour chaque observation
3
Déterminer si c'est un core point
4
Créer les clusters

Les hyperparamètres du modèle

- **Minimum Sample ⇛** Nombre minimal d'observations voisines pour qu'un point soit considéré « core »

- **Epsilon ⇛** Rayon dans lequel on considère qu'un point est « voisin »

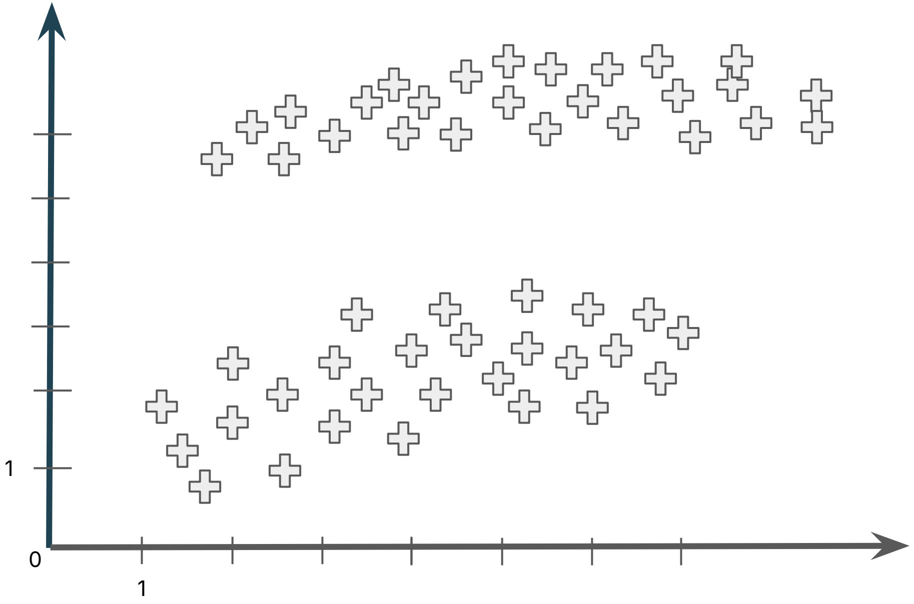
Example - Define min_sample & eps



min_sample = 4
eps = 0.2

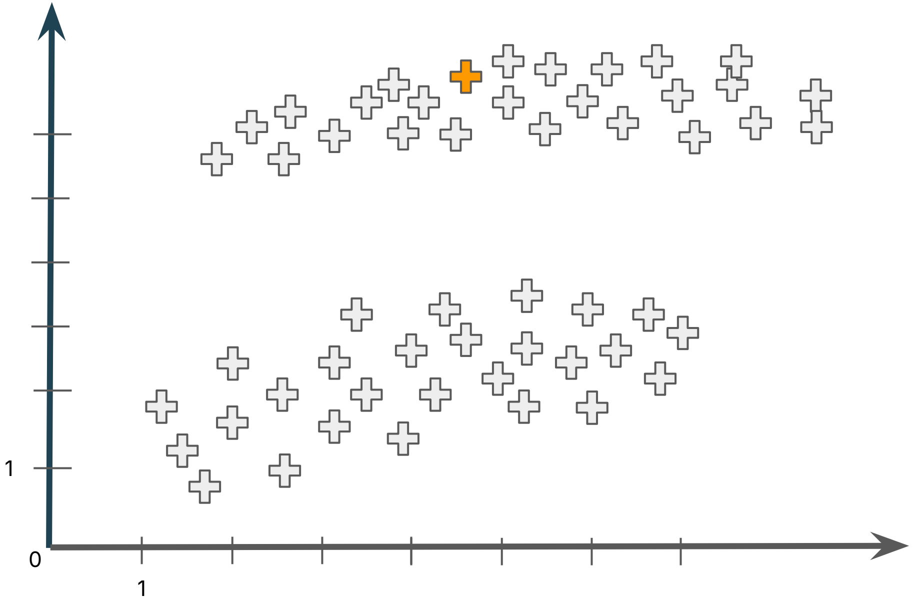
Example - Take observation & define core samples

min_sample = 4
eps = 0.2



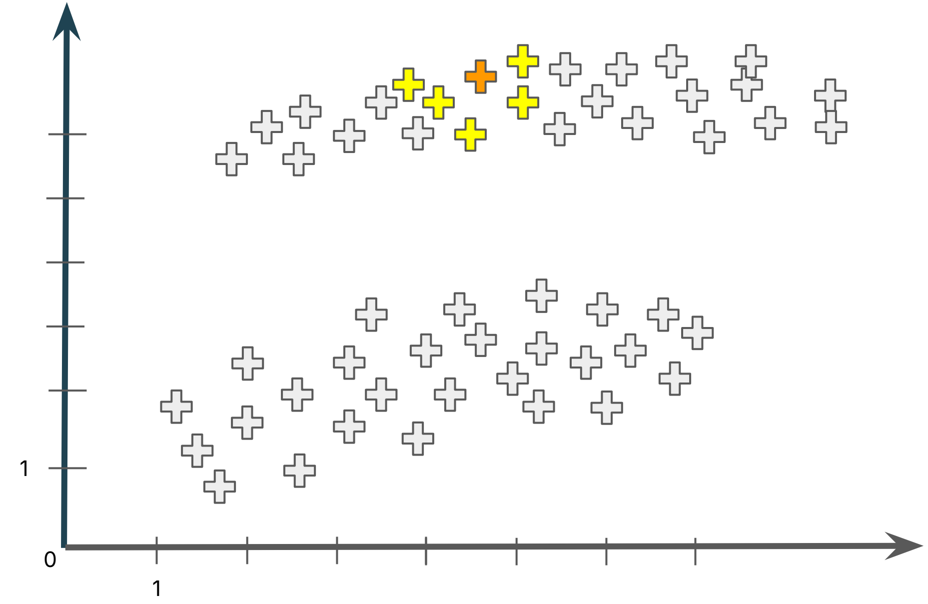
Example - Take observation & define core samples

min_sample = 4
eps = 0.2



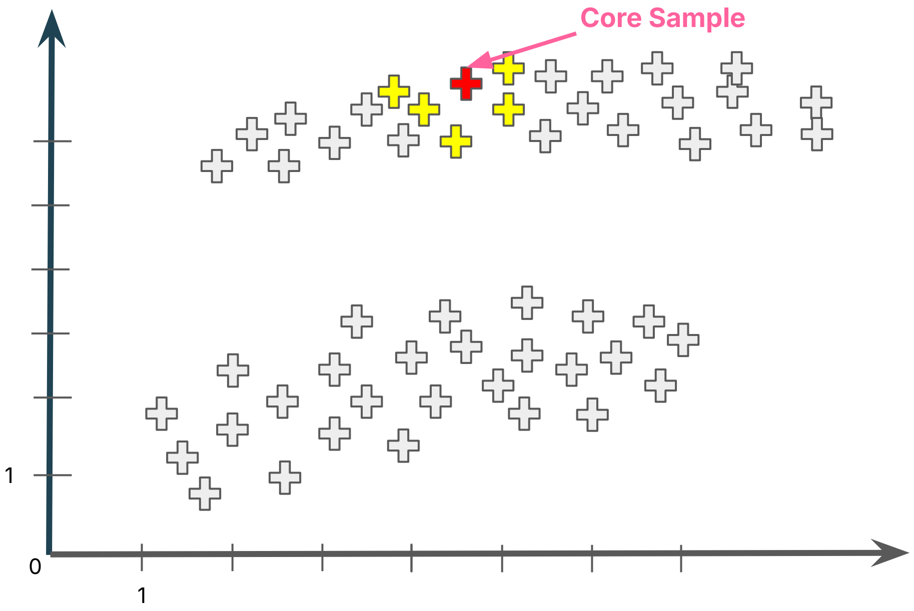
Example - Take observation & define core samples

min_sample = 4
eps = 0.2



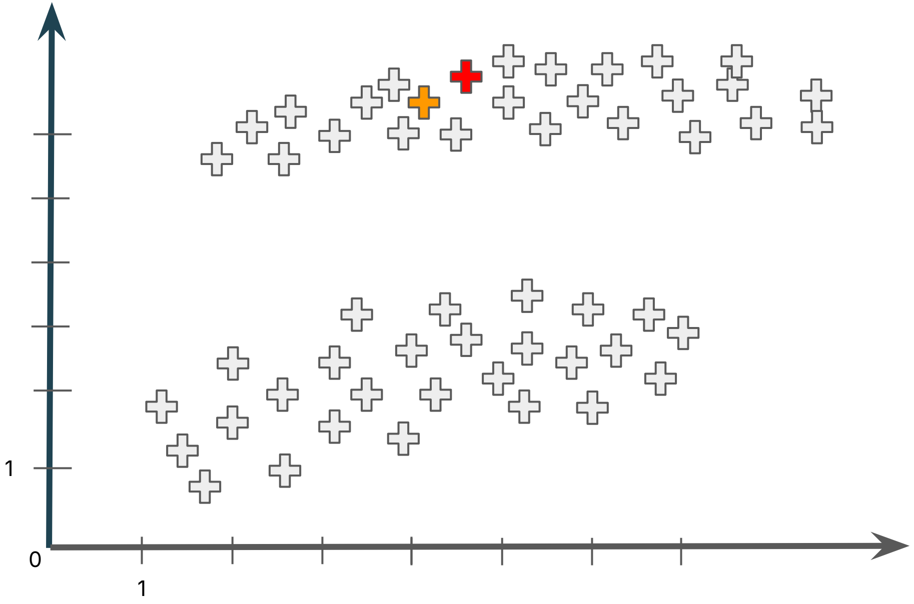
Example - Take observation & define core samples

min_sample = 4
eps = 0.2



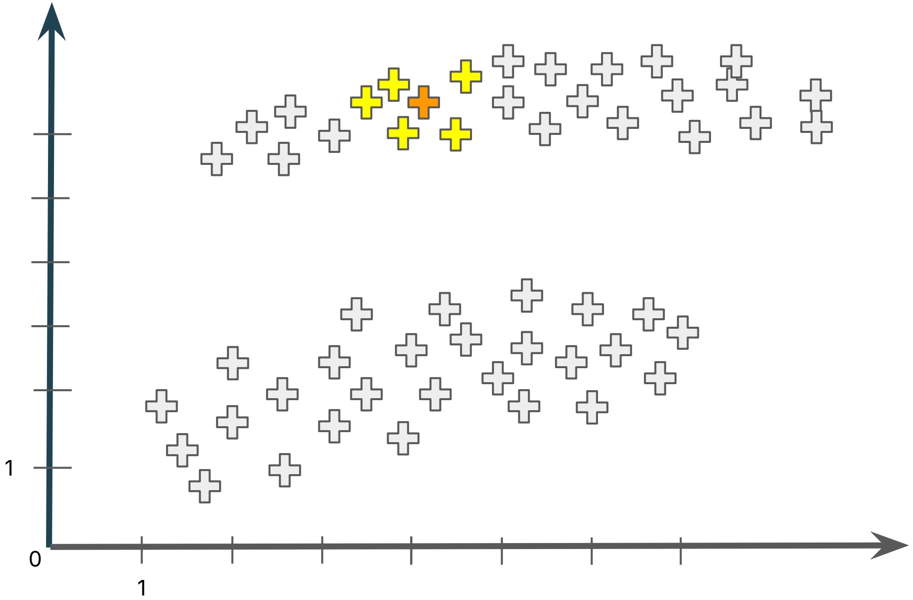
Example - Take observation & define core samples

min_sample = 4
eps = 0.2

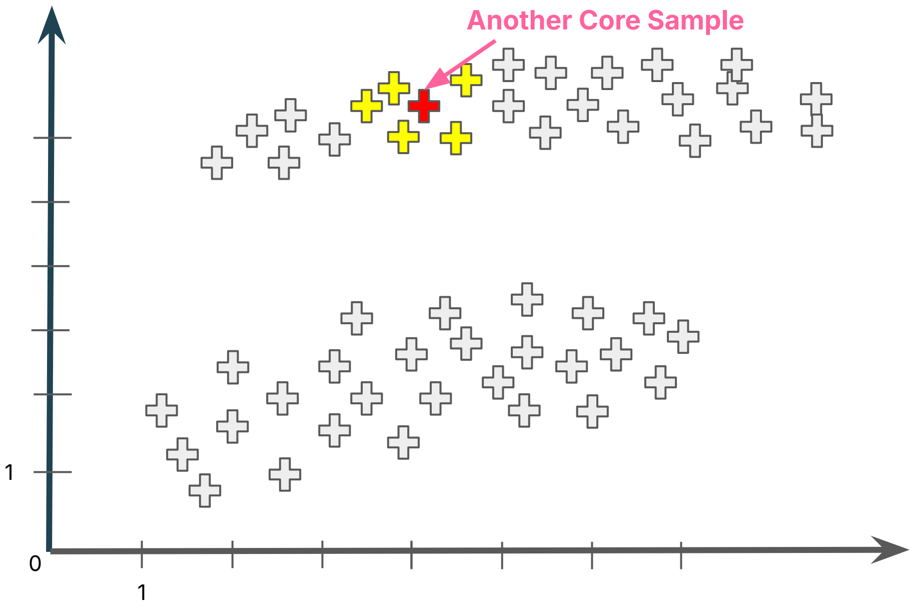


Example - Take observation & define core samples

min_sample = 4
eps = 0.2



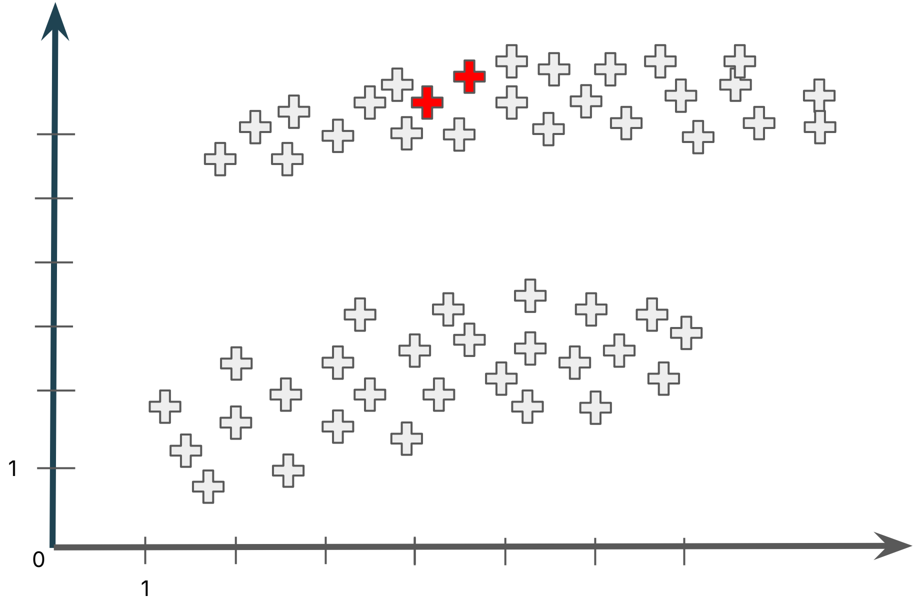
Example - Take observation & define core samples



min_sample = 4
eps = 0.2

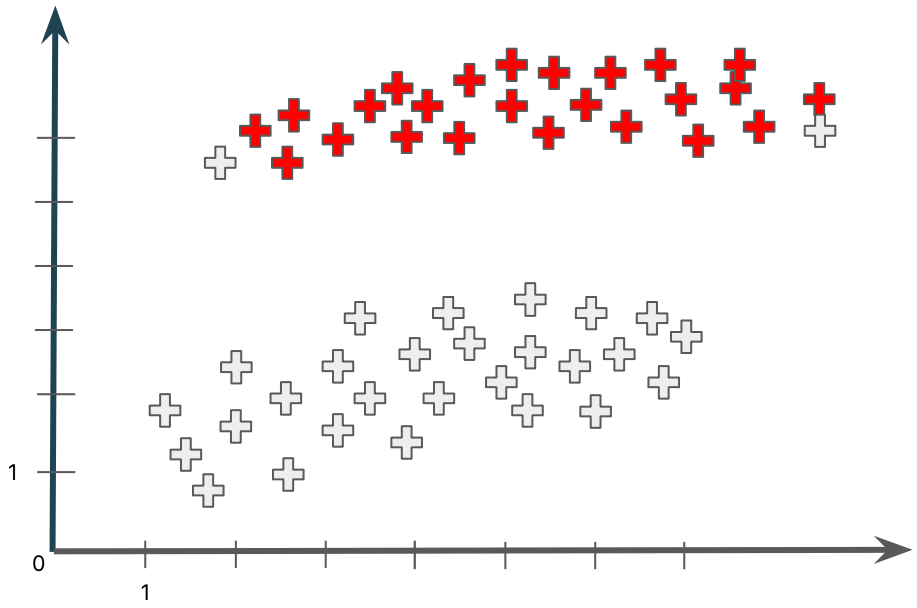
Example - Take observation & define core samples

min_sample = 4
eps = 0.2



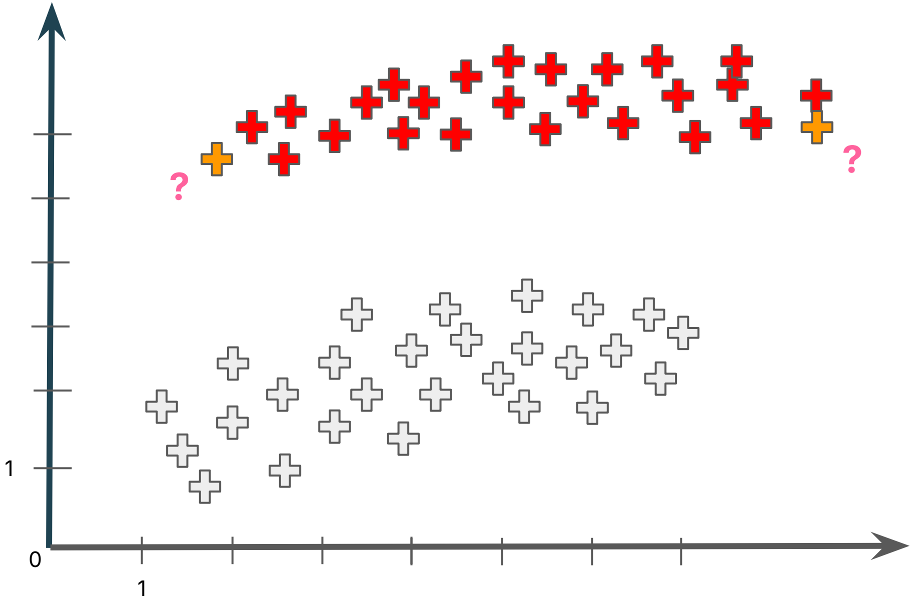
Example - Fast Forward

min_sample = 4
eps = 0.2



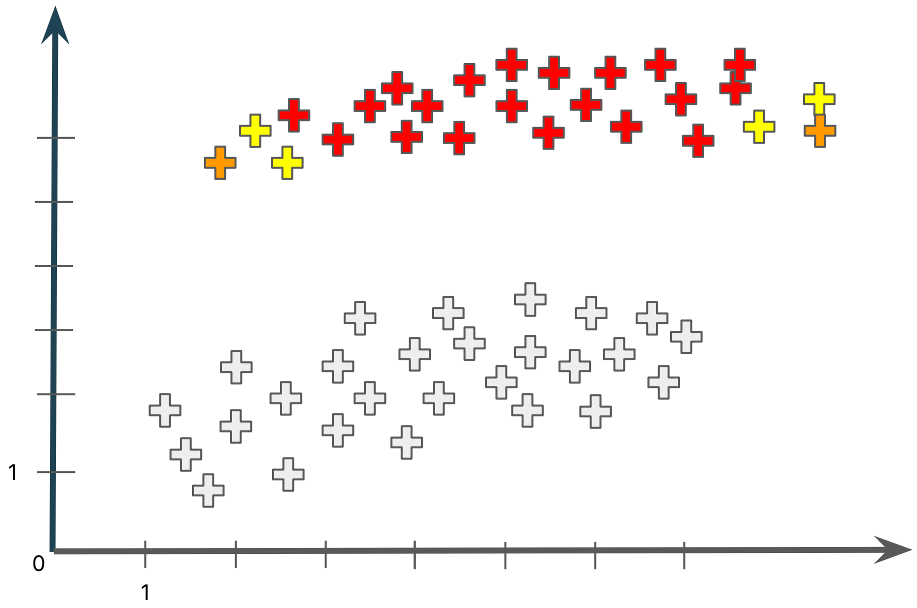
Example - Non-core samples

min_sample = 4
eps = 0.2



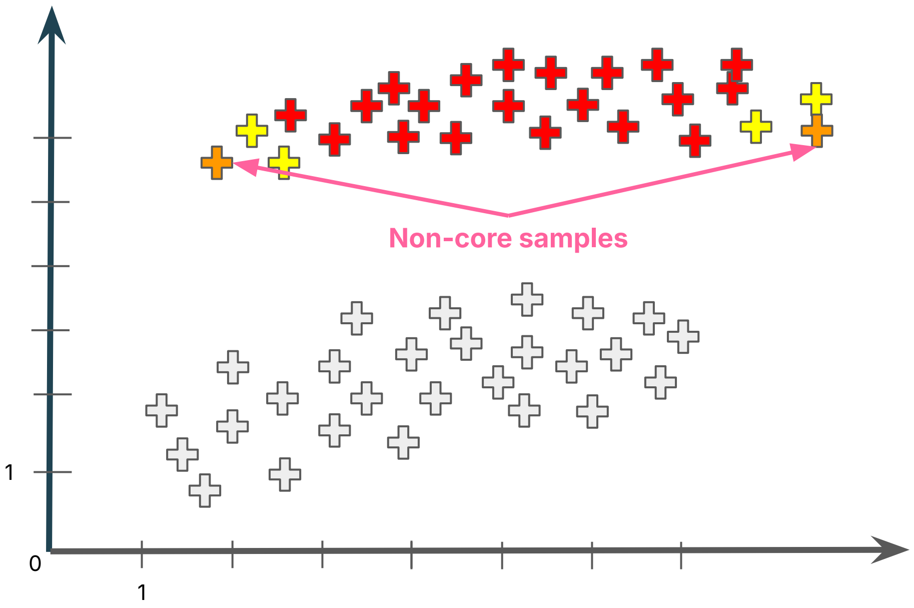
Example - Non-core samples

min_sample = 4
eps = 0.2



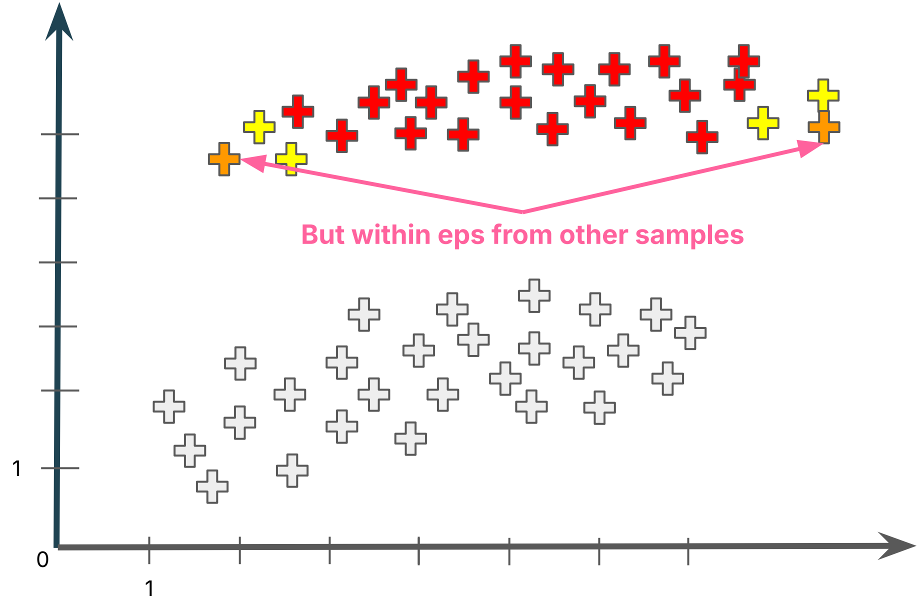
Example - Non-core samples

min_sample = 4
eps = 0.2

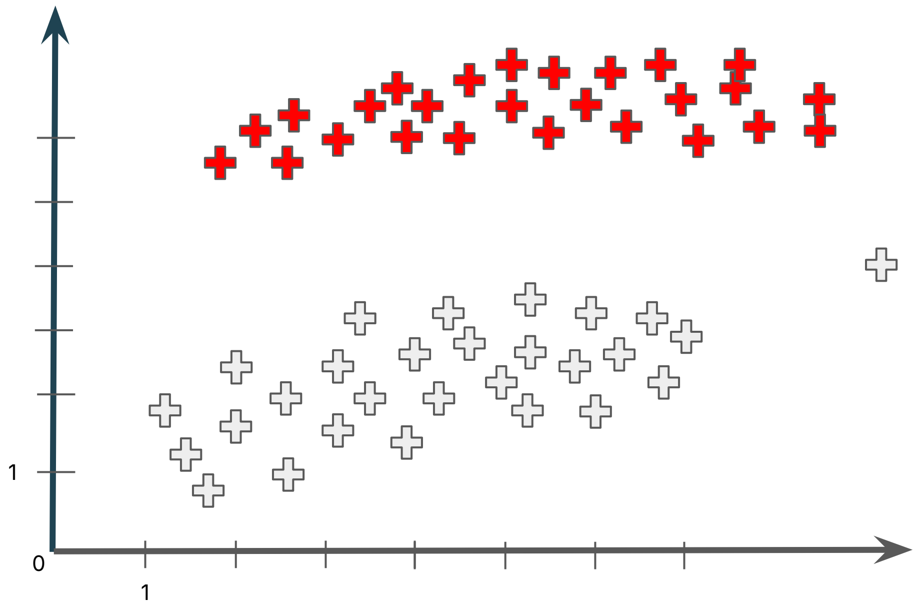


Example - Non-core samples

min_sample = 4
eps = 0.2

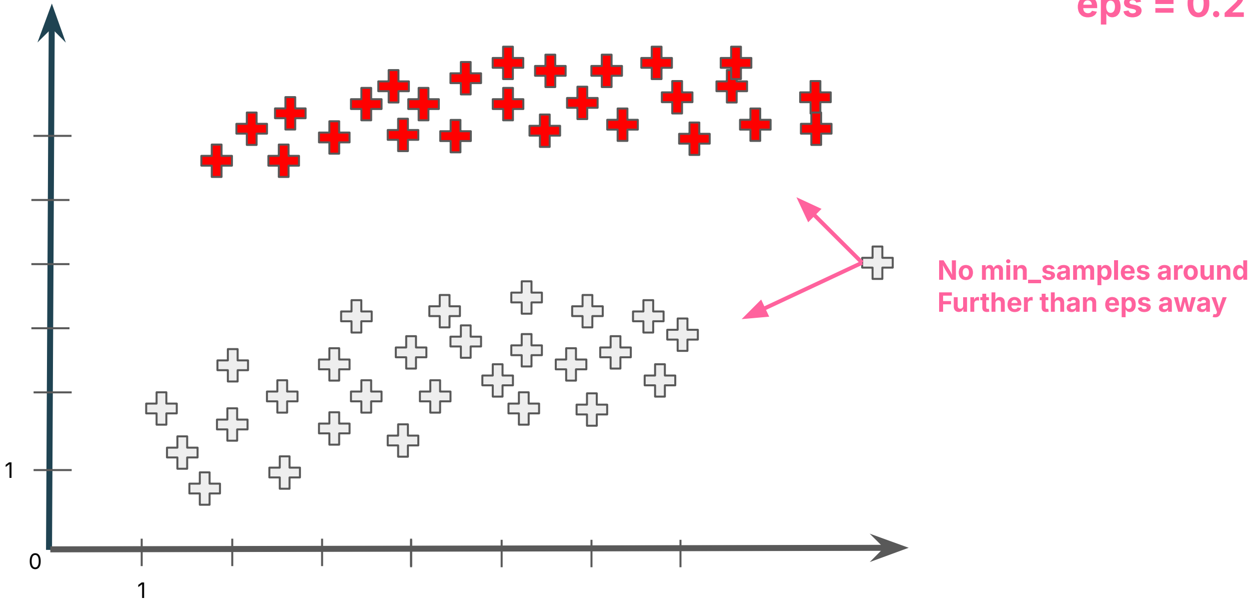


Example - Define outliers



min_sample = 4
eps = 0.2

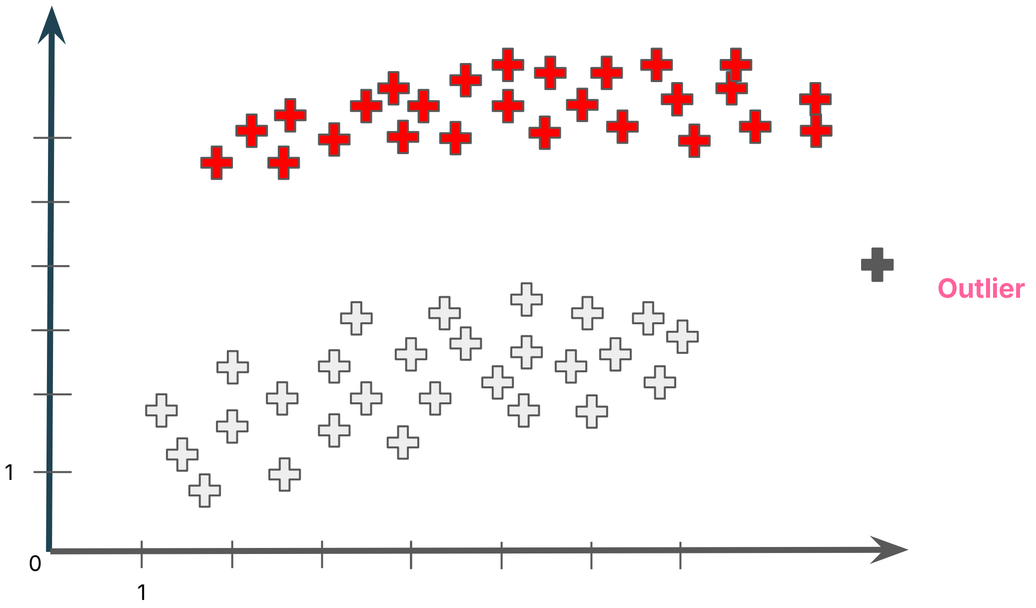
Example - Define outliers



min_sample = 4
eps = 0.2

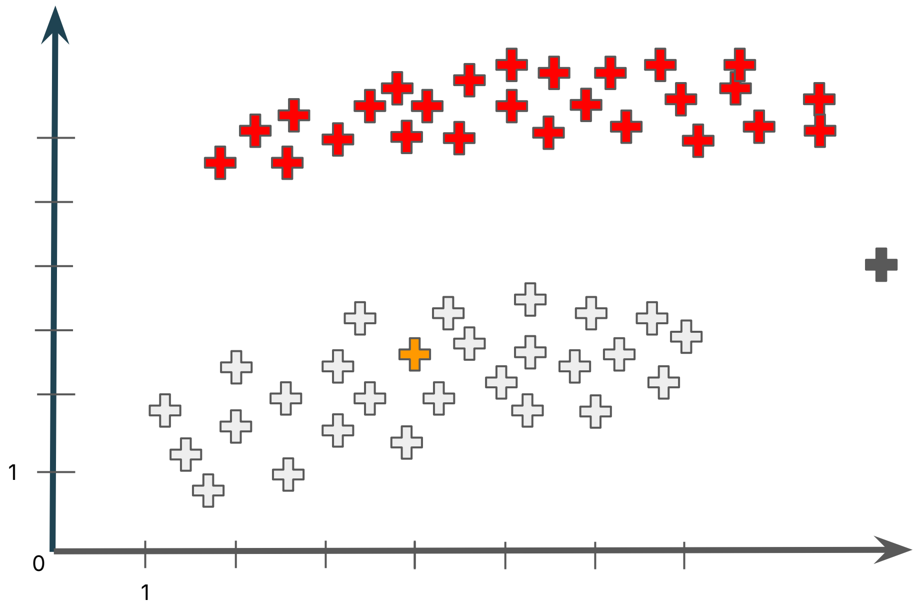
Example - Define outliers

min_sample = 4
eps = 0.2



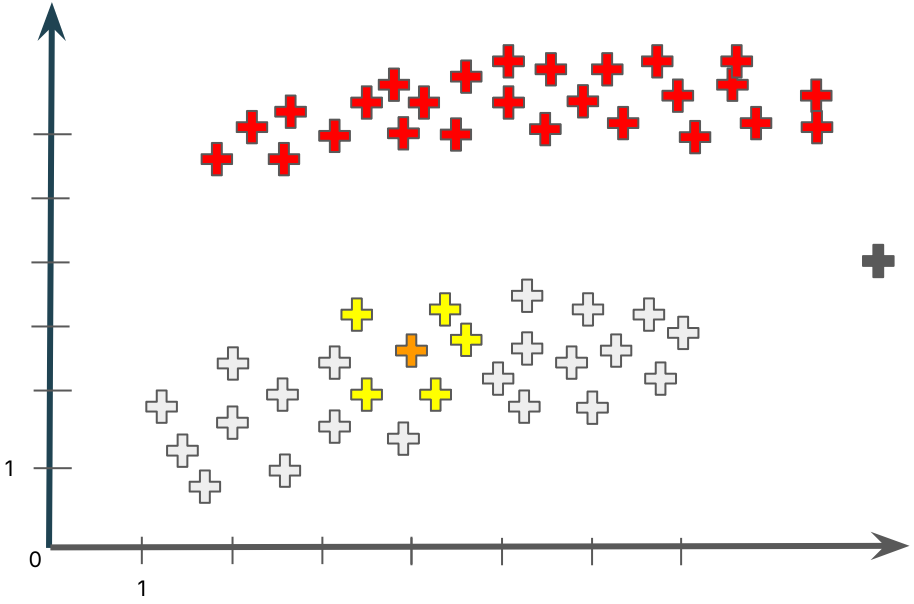
Example - Same process with second cluster

min_sample = 4
eps = 0.2



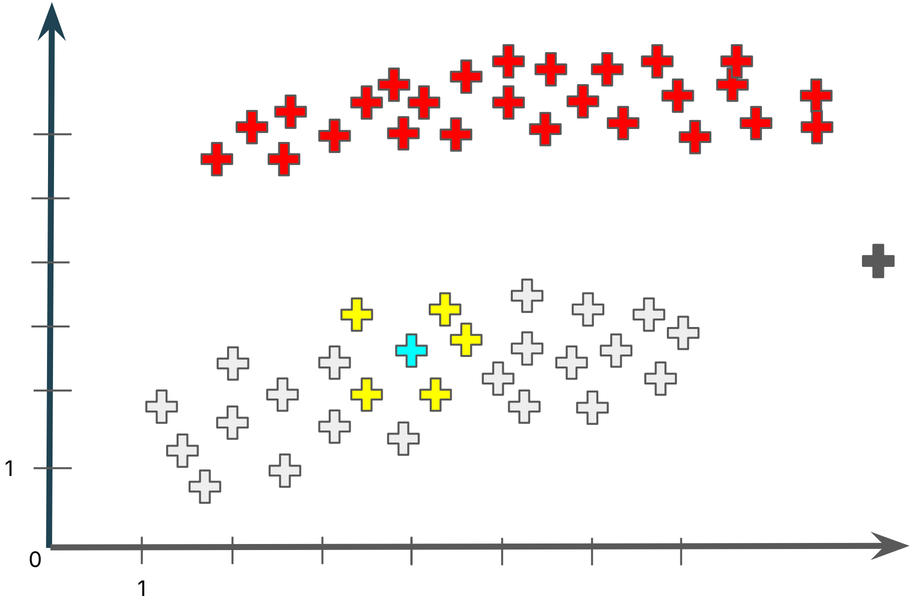
Example - Same process with second cluster

min_sample = 4
eps = 0.2



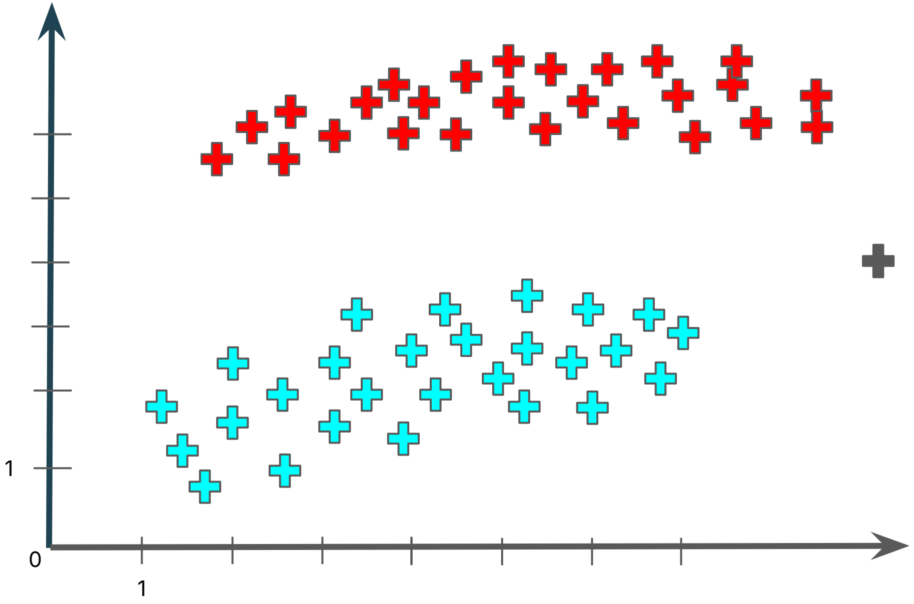
Example - Same process with second cluster

min_sample = 4
eps = 0.2



Example - Same process with second cluster

min_sample = 4
eps = 0.2



Comment choisir min_sample & eps?

How to set min_sample & eps?

- **Low eps & Low min_sample** → High outliers sensitivity
- **High eps & High min_sample** → Low outliers sensitivity

How to set min_sample & eps?

- **Low eps & High min_sample** → High density clusters
- **High eps & Low min_sample** → Low density clusters
- **Low eps & Low min_sample** → High outliers sensitivity
- **High eps & High min_sample** → Low outliers sensitivity

Data Science Bootcamp

Questions

Jedha