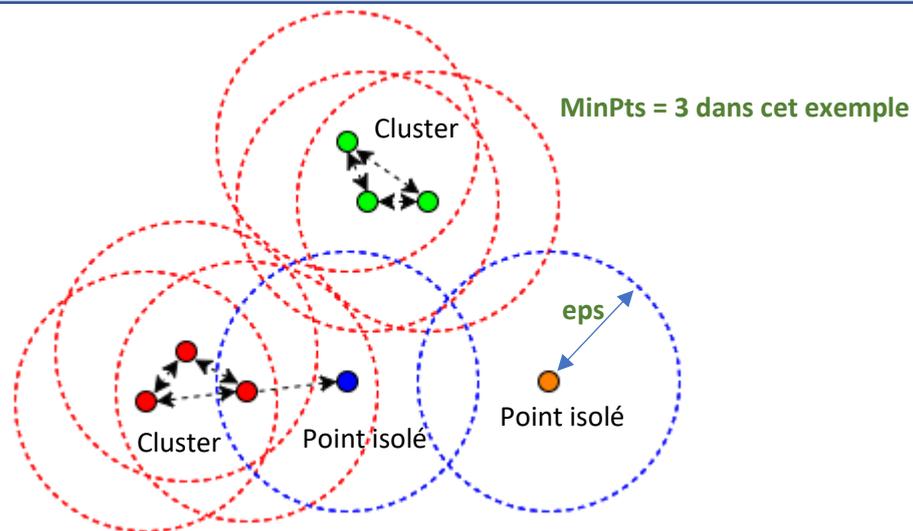


DBSCAN

L'algorithme DBSCAN est un algorithme d'apprentissage **non-supervisé** qui crée des clusters basés sur une notion de **densité**. Quand des points sont groupés, l'algorithme va créer un cluster et chercher tous les points suffisamment proches pour les ajouter.



Algorithme :

Entrées :

D : L'ensemble des points
 MinPts : Nombre minimal de points pour créer un cluster
 eps : Distance maximale entre 2 points pour les considérer voisins

Processus :

```

C = 0
pour chaque point P non visité des données D
  marquer P comme visité
  PtsVoisins = epsilonVoisinage(D, P, eps)
  si tailleDe(PtsVoisins) < MinPts
    marquer P comme BRUIT
  sinon
    C++
    etendreCluster(D, P, PtsVoisins, C, eps, MinPts)
  retourner C
  
```

Sortie :

C : Les clusters (liste de liste) avec leurs points respectifs

Fonctions supplémentaires :

```

etendreCluster(D, P, PtsVoisins, C, eps, MinPts) Etend un cluster
ajouter P au cluster C
pour chaque point P' de PtsVoisins
  si P' n'a pas été visité
    marquer P' comme visité
    PtsVoisins' = epsilonVoisinage(D, P', eps)
    si tailleDe(PtsVoisins') >= MinPts
      PtsVoisins = PtsVoisins U PtsVoisins'
  si P' n'est membre d'aucun cluster
    ajouter P' au cluster C
  
```

```

epsilonVoisinage(D, P, eps) Retourne les voisins d'un point
retourner tous les points de D qui sont à une distance
inférieure à epsilon (eps) de P
  
```

Avantages :

- Permet de créer des clusters non-convexes
- Gère les données anormales (bruit)

Inconvénients :

- Des paramètres difficiles à définir (epsilon, minPts)

Paramètres importants :

MinPts : Nombre minimal de points pour créer un cluster

Eps (Epsilon) : Distance maximale entre 2 points pour les considérer comme voisins