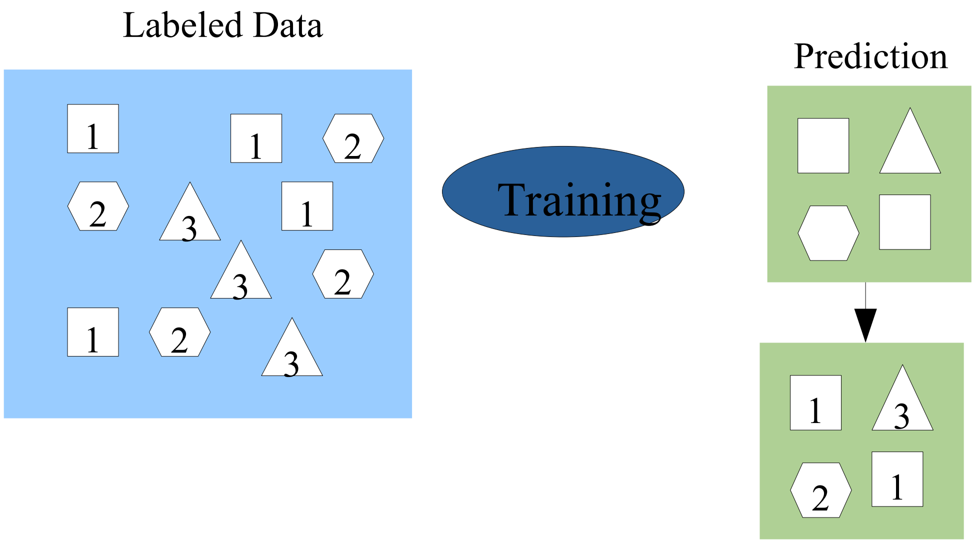
Introduction

Supervised Learning

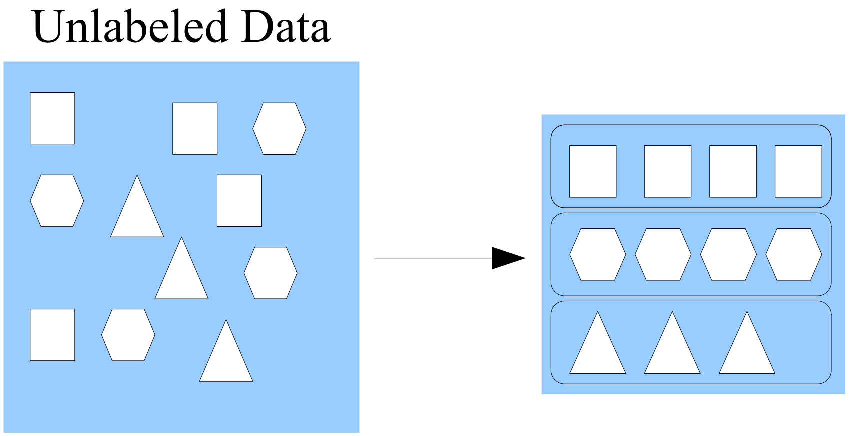
- **Map input data** with **the output data**.
- Supervision, as student learns in the supervision of the teacher.
- Example : spam filtering
- Can be grouped in two categories :
  - Classification
  - Regression



Unsupervised Learning

- Data **not labelled**, classified, or categorized
- The algorithm needs to **act** on that data **without any supervision**.
- Can be grouped in two categories :
  - Clustering
  - Dimensionality Reduction

reduction de nombre de colonne
viusalisation de donnée sur (x,y)



Introduction

Unsupervised Learning

Dimensionality Reduction

Conducted to reduce the variable space before analysis

- PCA : Principal Component Analysis m dim ->n dim
- t-NSE : t-distributed Stochastic Neighbor Embedding plus efficace reduction de dim a 2 ou 3 colonne seulement
- Neural Networks

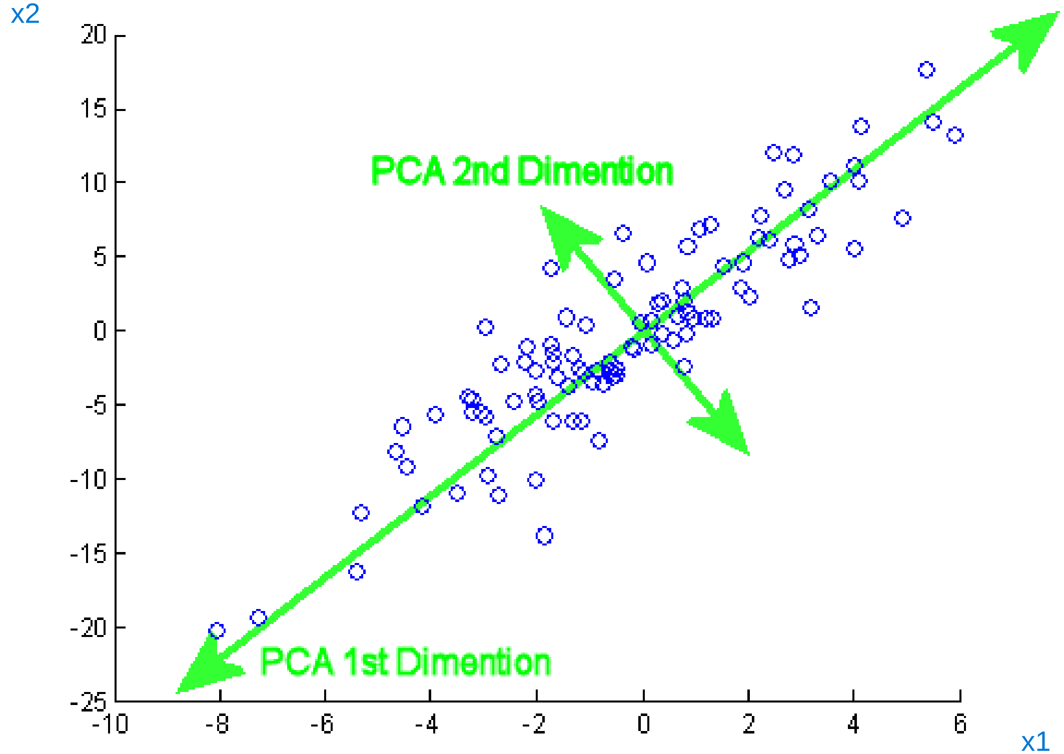
Clustering

Discover groupings inside the input data

- K-means clustering
- DBSCAN : Density-based spatial clustering of applications with noise
- Isolation Forest(Anomaly) trouver des données anormal
- Neural Networks

Dimensionality Reduction

PCA : Principal Component Analysis



on voit que x1 est suffisant
on peut faire une projection
sur une seul axe

Creating new features that are linear combinations of the original features in the dataset.

Dimensionality Reduction

PCA : Principal Component Analysis

Step 1 : Standardization $z = \dfrac{value - mean}{standard\ deviation}$

Step 2 : Compute Covariance Matrix $\sigma_{xy} = \text{cov}(x,y) = \dfrac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

n var. -> matrix de covariance-> correlation entre les cols-> n val propre du matrice de Cov

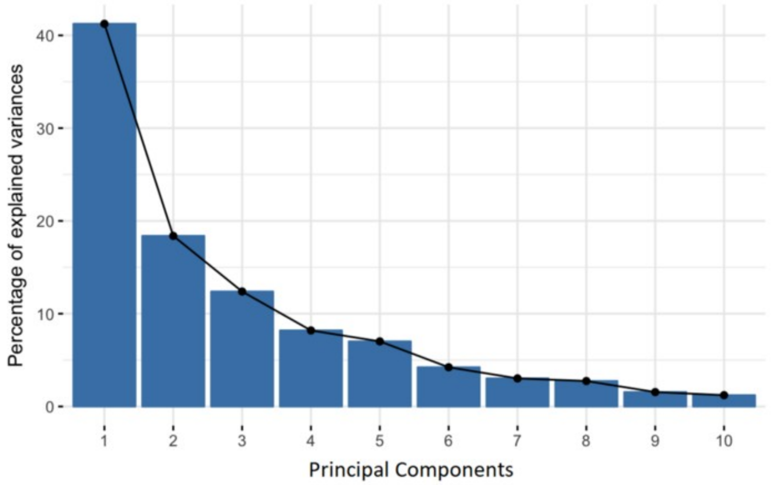
- if >0 : the two variables increase or decrease together (correlated)
- if <0 : One increases when the other decreases (Inversely correlated)

comment choisir la val de reduction-> les cols les plus correlés

Step 2 : Compute eigenvectors & eigenvalues to identify the principal components

Xi = di/ sum(lambda(i)) sum(lambda(i) / sum(lambda(j) = 95 % c'est bon

New variables with :
Max possible information in the first component, then maximum remaining information in the second and so on.

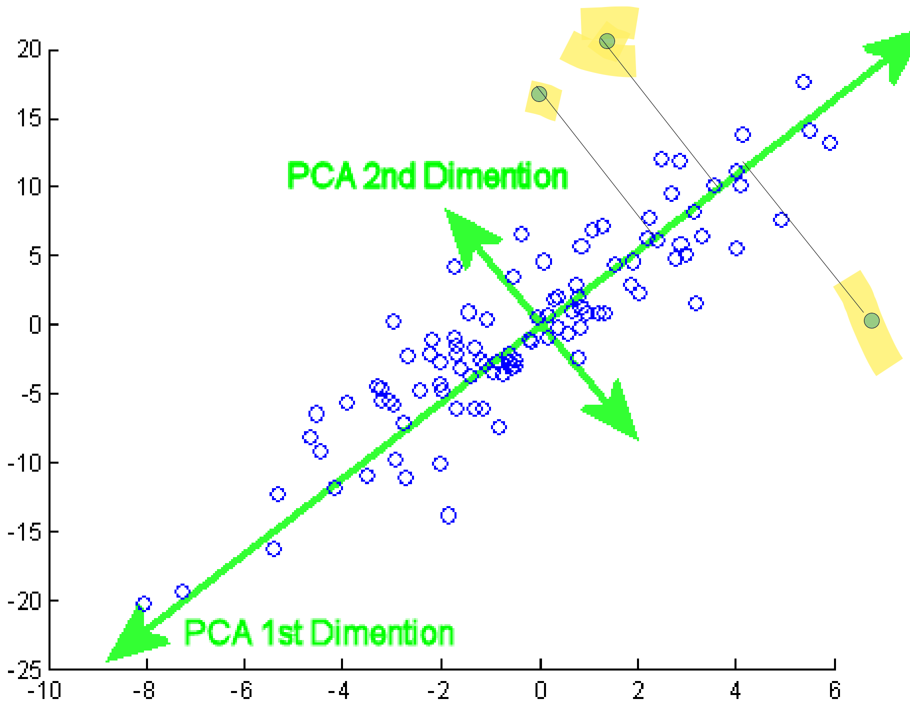


Dimensionality Reduction

t-NSE : t-distributed Stochastic Neighbor Embedding

visualisation, grande temps de calcul par rapport a PCA

PCA tries to preserve the Global Structure of data i.e when converting d-dimensional data to d'-dimensional data then it tries to map all the clusters as a whole due to which local structures might get lost.



information sur la localité n'est pas concervé
mais ters pratique, car temps de calcul est 100x meilleur

Dimensionality Reduction

t-NSE : t-distributed Stochastic Neighbor Embedding

- It embeds the points from a higher dimension to a lower dimension trying to preserve the neighborhood of that point.

- Informally, the algorithm places randomly all points on the *2D plane*, and lets them interact as if they were physical particles.

- The interaction is governed by two laws: first, all points are repelled from each other; second, each point is attracted to its nearest neighbours choix des interaction: potentielles de repulstion& attraction

- The most important parameter of t-SNE, called perplexity, determines how many of its nearest neighbours each point is attracted to.

perplexity= proba de presence autour d'un point

Dimensionality Reduction

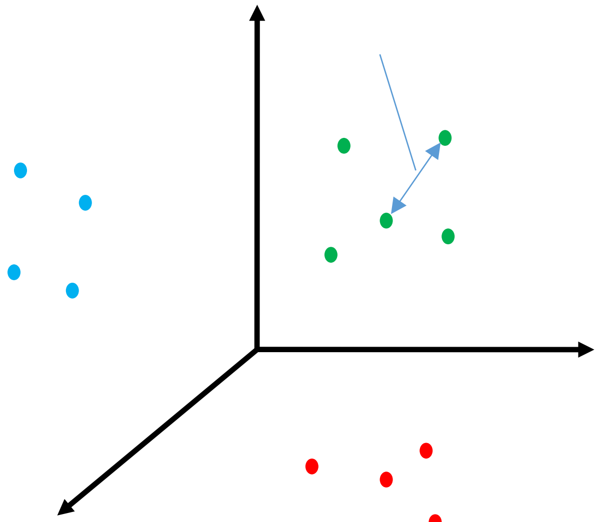
t-NSE : t-distributed Stochastic Neighbour Embedding

Perplexity $$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

$$q_{ij} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l\neq k}(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

Gaussian

Student t-distribution



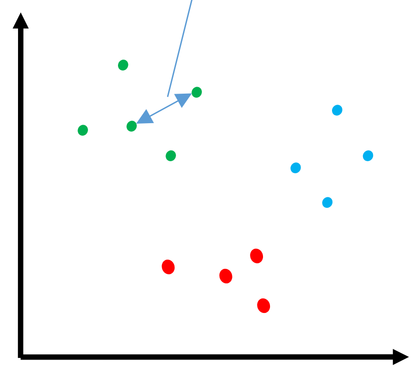
$x_i \in \mathbb{R}^N$

t-SNE

Optimize

$$C = \sum_i \sum_j p_{i|j} \log \frac{p_{i|j}}{q_{i|j}}$$

revient a un probelem de minimization de C



$y_i \in \mathbb{R}^d \; where \; d < N$

Goal : find the y_i through optimization such that p and q are as close together as possible

Maaten & Hinton, JMLR, 2008

Clustering

K-means

random choice of k cluster centers

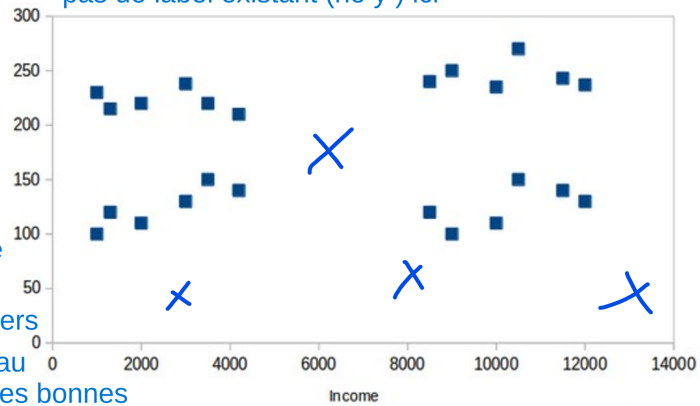
tout point affecter a un centre

recalcul du centre avec les clusters

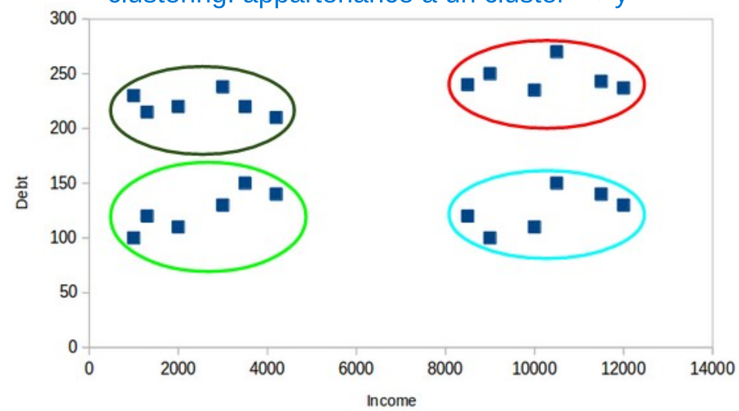
placement du du centre a nouveau

enfin les centre ne bouge plus> les bonnes

pas de label existant (no y) ici



clustering: appartenance a un cluster = >y



Step 1: Choose the number of clusters k hyperparameter

Step 2: Select k random positions for centroids

Step 3: Assign all the points to the closest cluster centroid

Step 4: Recompute the centroids of newly formed clusters

Step 5: Repeat steps 3 and 4

Stopping criteria :

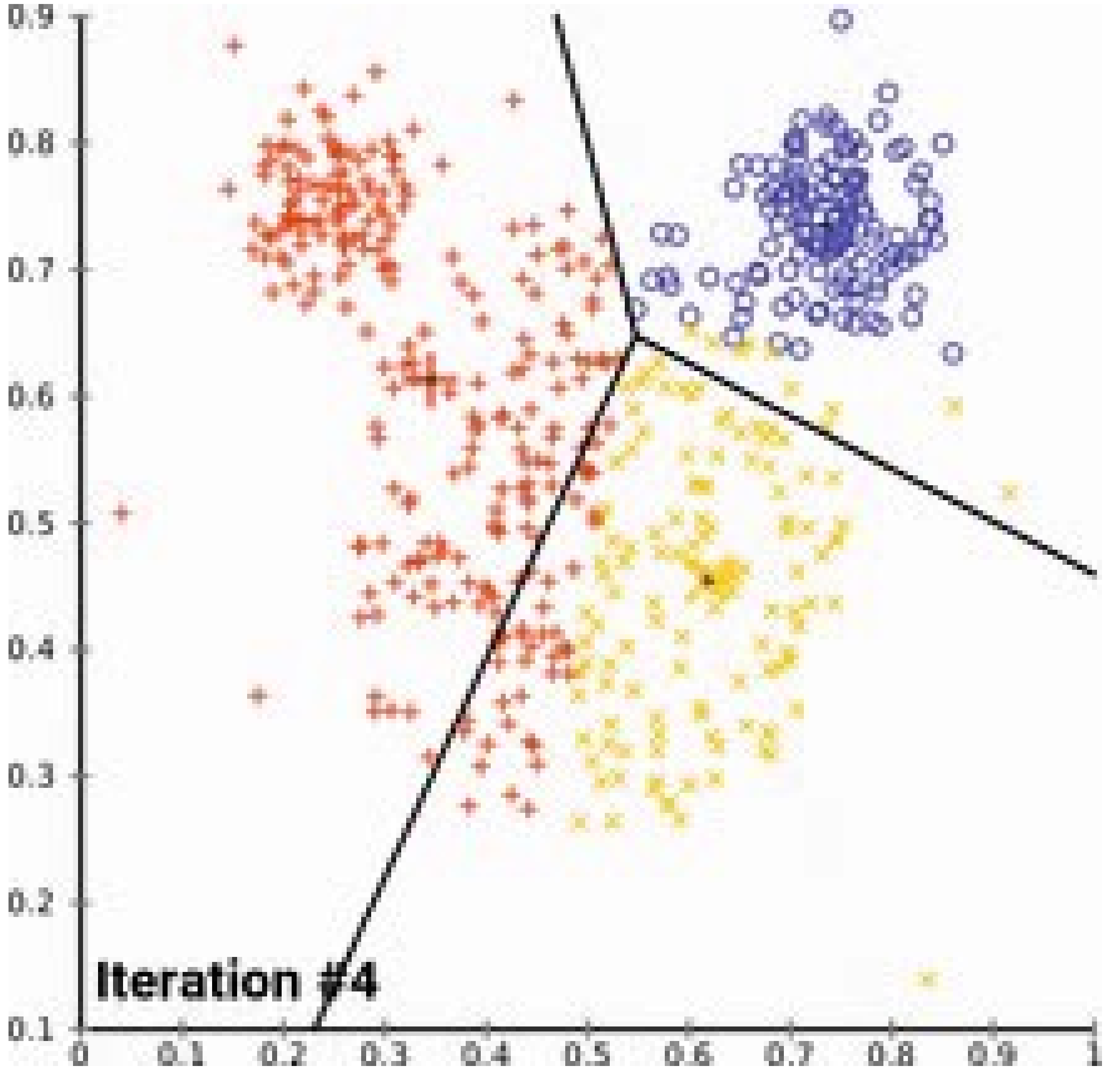
Centroids of newly formed clusters do not change

Points remain in the same cluster

Maximum number of iterations are reached

Clustering

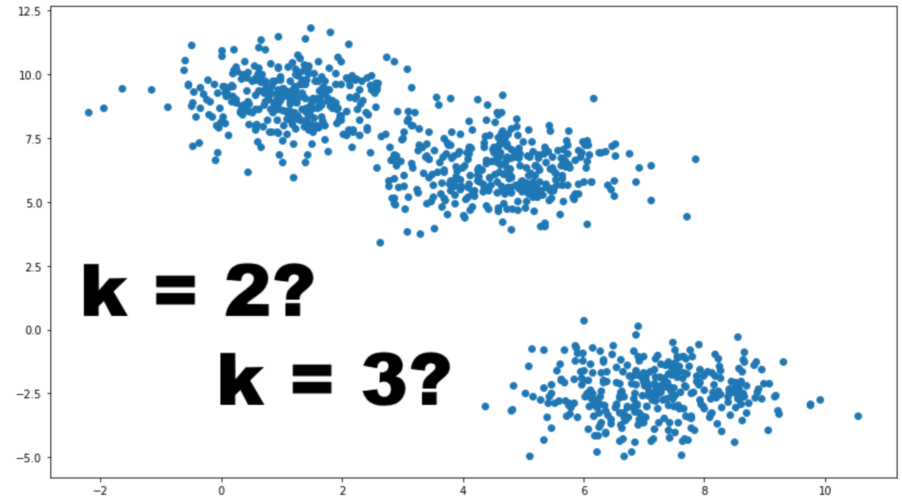
K-means



Clustering

K-means

How to Determine the Optimal K ?

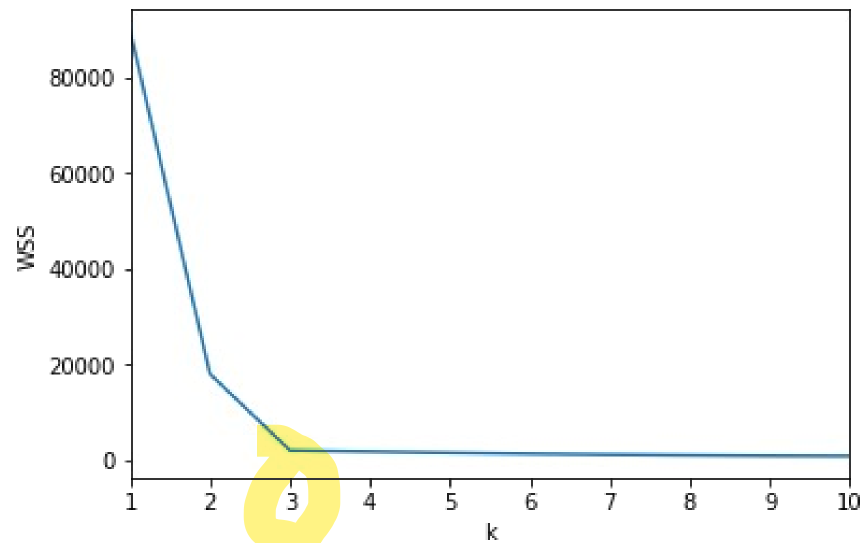


Calculate the Within-Cluster-Sum of Squared Errors (**WSS**) for different values of k.

Choose the k for which WSS becomes first starts to diminish.

(Elbow Method)

Wss = sum(dist.AuCentre^2)
Wss diminue
lorsque Wss ne diminue pas trop-> n_cluster est bonne



Clustering

DBSCAN : Density-based spatial clustering of applications with noise

• DBSCAN uses a minimal distance **ε** and a minimal number, **minPts,** of cluster seed points as input

• It finds dense regions in the data and uses them as cluster seeds. Then it expands these clusters and adds points if they are within distance to a cluster. All points not within reach of a cluster are marked as outliers.

• DBSCAN can deal with clusters of arbitrary shapes.

• Distance based clustering are difficult in high dimensions (curse of dimensionality)

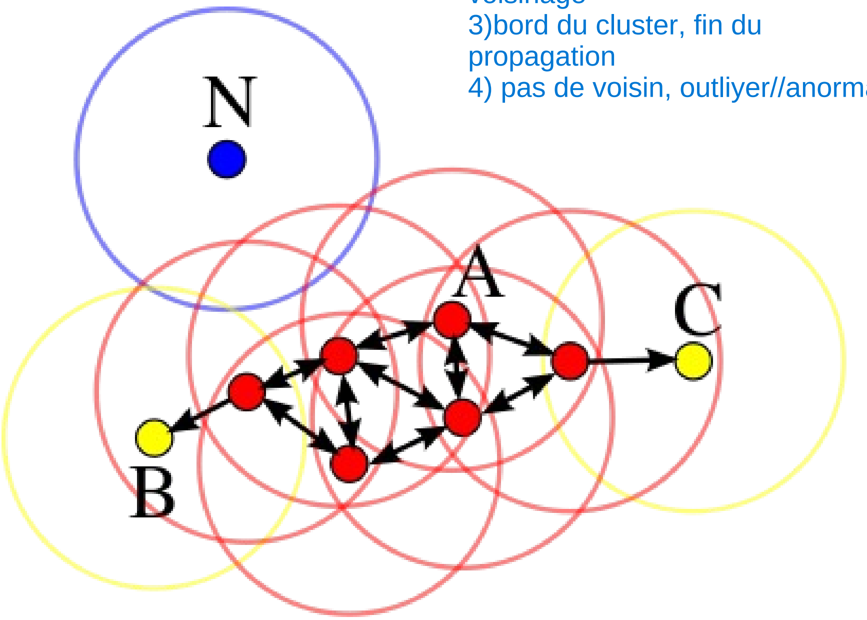
Clustering

DBSCAN : Density-based spatial clustering of applications with noise

1)depart d'un point ou la densité de point est grand
2)nombre de point dans le voisinage
3)bord du cluster, fin du propagation
4) pas de voisin, outliyer//anormal

distance entre voisin

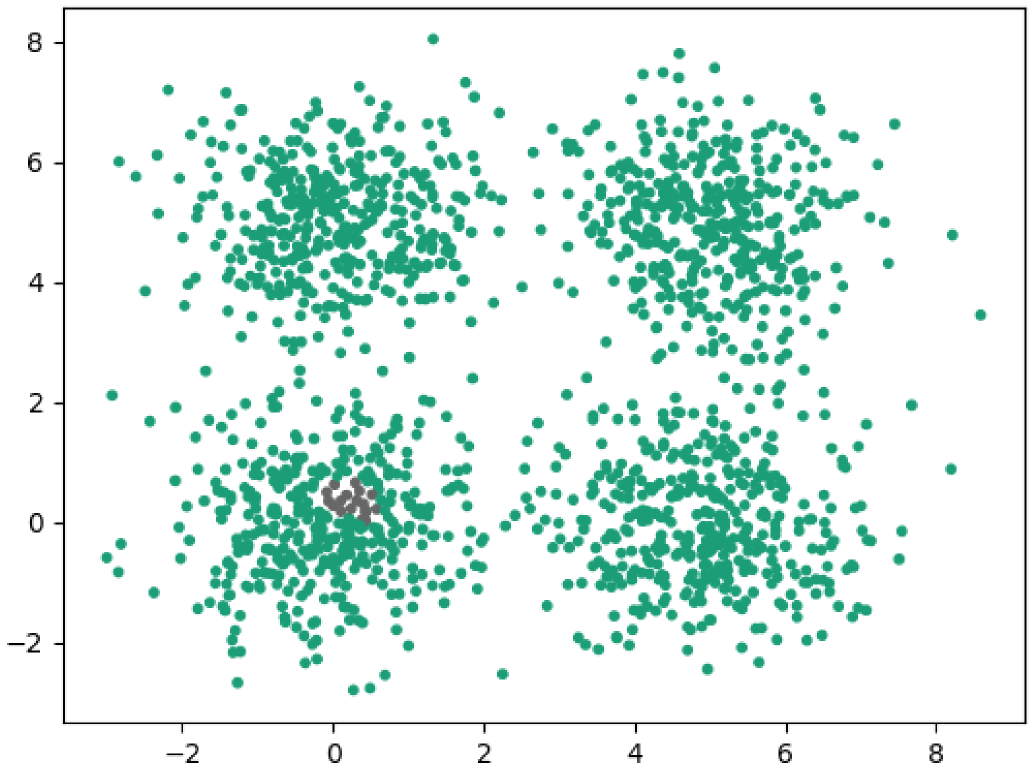
- Minimal distance **ε** & minimal number, **minPts,** of cluster seed points.
- Dense regions as cluster seeds. Expands these clusters. Points not in a cluster are marked as outliers.
- Can deal arbitrary shapes.
- Difficult in high dimensions.

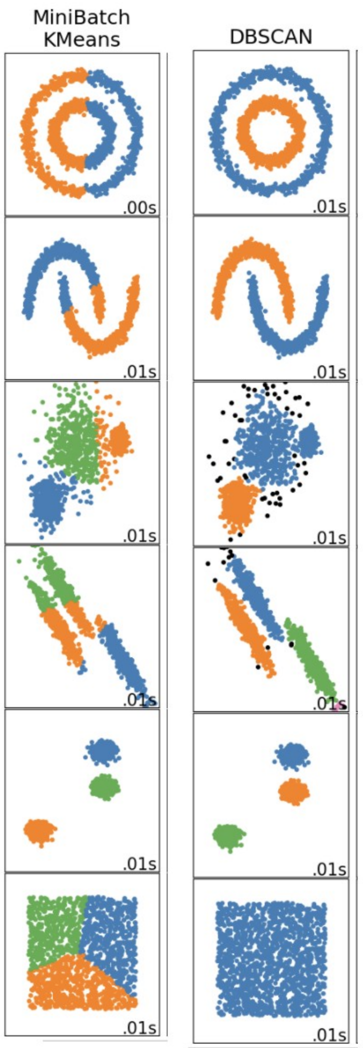


In this diagram, **minPts** = 4.
Point A and the other red points are **core** points (points in an **ε** radius contain at least 4 points (including the point itself).
B and C are not core points, but are **border** points
Point N is a **noise** point that is neither a core point

Clustering

DBSCAN : Density-based spatial clustering of applications with noise





Clustering

Isolation Forest

Isolation forest is an anomaly detection algorithm.

Isolation Forest can detect anomalies **faster** and we require **less memory** compared to other algorithms.

Anomaly detection is commonly used for:

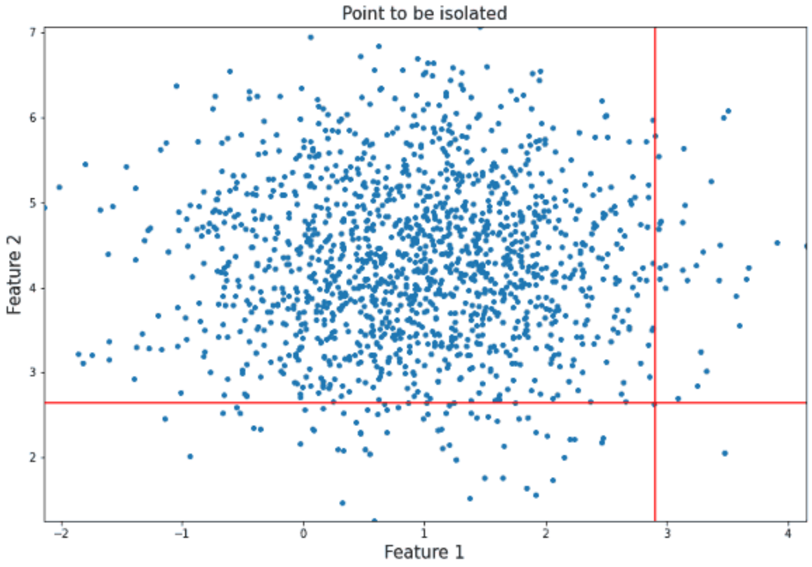
- Data cleaning
- Intrusion detection
- Fraud detection
- Systems health monitoring

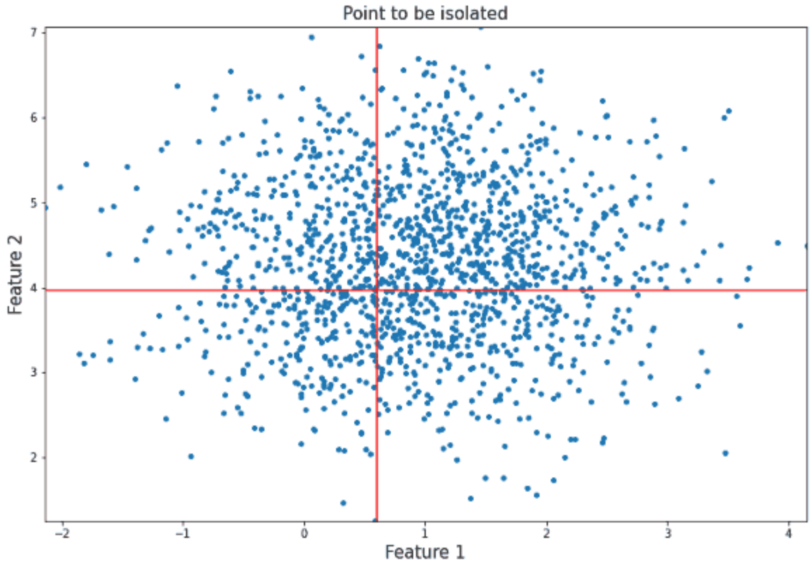
Clustering

Isolation Forest

isoler un point qui est tout seul

- Isolation Forest is based on the **Decision Tree** algorithm.
- **Randomly** selecting a **feature** + **randomly** selecting a **split** value between the max and min values.
- The random partitioning will produce **shorter paths** in trees for the **anomalous** data

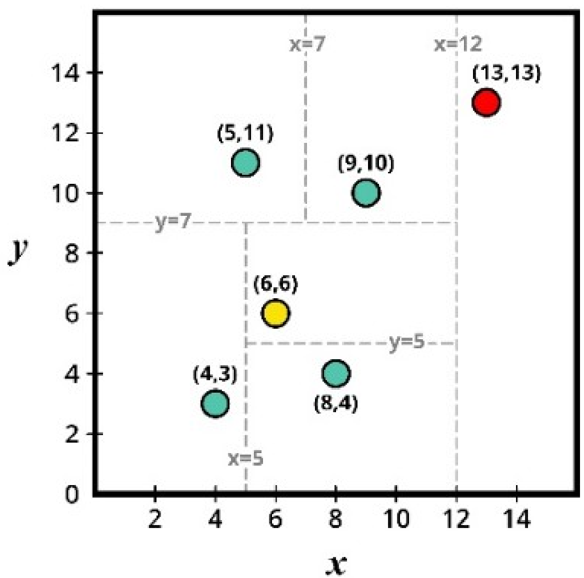


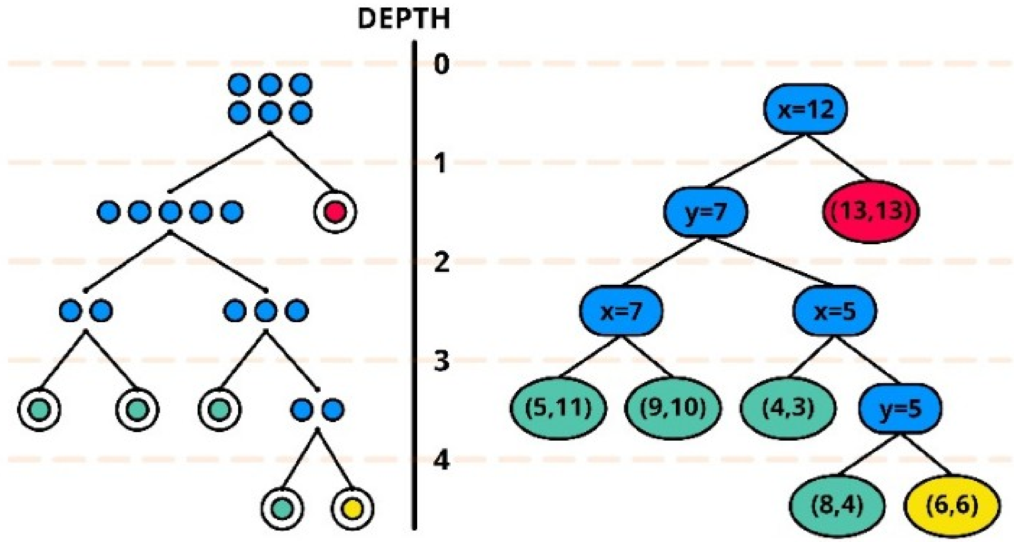


Clustering

Isolation Forest

DEC: le decoupe est au hasard: le nombre de fois ou on applique le decoupage est un hyperparamter a entrer, pour que l'algorithm prend fin a un moment





Hyperparameters :

The number of base estimators in the **ensemble**. (DEC)

The number of samples to draw from X to train each base estimator

The amount of contamination of the data set : percentage of anomaly

The number of features to draw from X