Intelligence Artificielle 2

Cours 7 - Algorithmes de ML - Non-supervisé

Table des matières

Définition

L'apprentissage non supervisé est un type d'algorithme d'apprentissage automatique ("Machine Learning") utilisé pour trouver les modèles, la structure ou la relation au sein d'un ensemble de données à l'aide de données non étiquetées.

Les problèmes non-supervisés sont **généralement plus difficiles** à traiter à cause du manque d'information respectif.

Définition

Un modèle non-supervisé va faire une prédiction, et renvoyer un résultat (une étiquette) tout comme un algorithme supervisé.

Pour combler le manque d'information puisqu'un modèle non-supervisé ne détient **aucune étiquette** dans son ensemble d'entraînement, celui-ci va généralement exploiter le principe de distance.

Définition

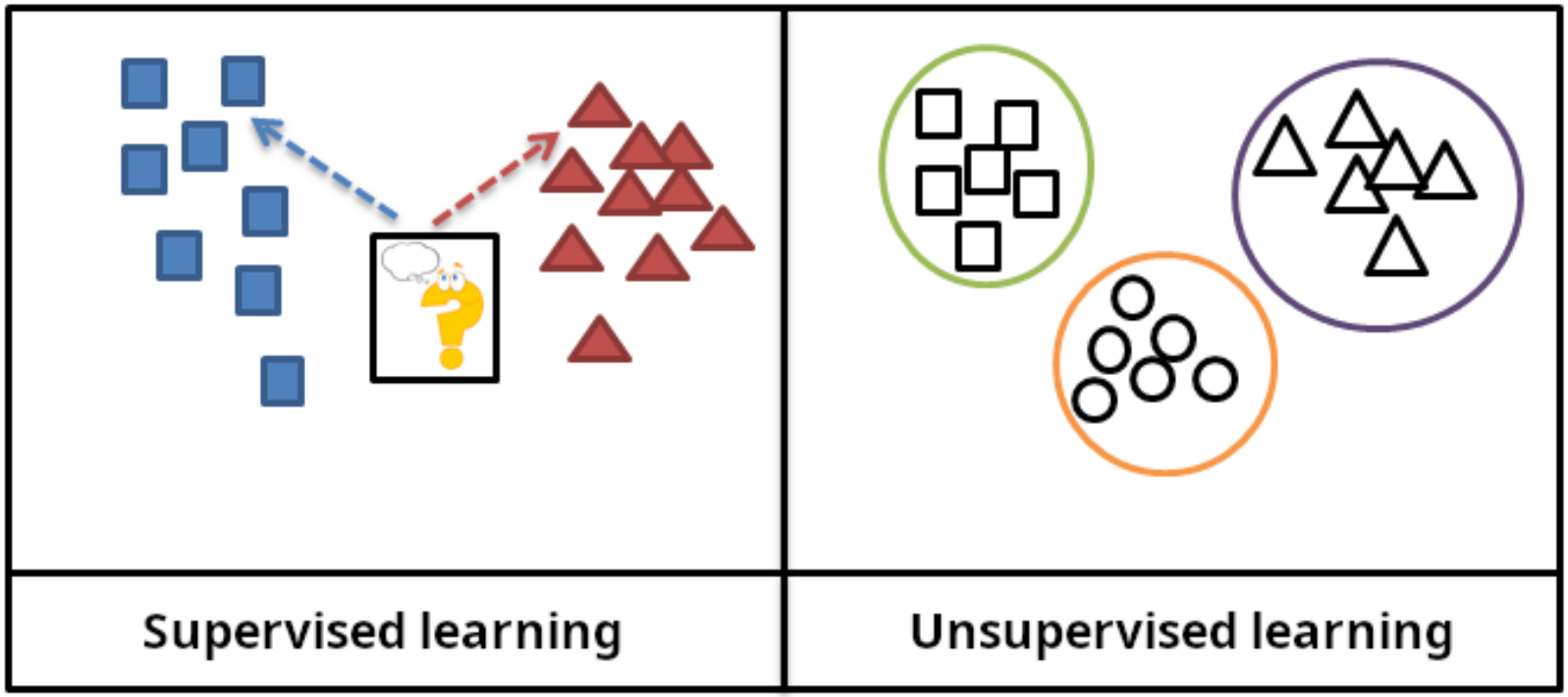

Figure: https://en.wikipedia.org/wiki/Supervised_learning

Non supervisé - Clustering

La tâche consistant à regrouper les points de données en fonction de leur similitude les uns avec les autres est appelée clustering ou analyse de cluster.

Cette méthode vise à obtenir des informations à partir de points de données non étiquetés, c'est-à-dire que, contrairement à l'apprentissage supervisé, nous n'avons pas de variable cible.

Non supervisé - Clustering

Le clustering vise à former des groupes de points de données homogènes à partir d'un ensemble de données hétérogènes.

Il évalue la similarité en fonction d'une métrique telle que la distance euclidienne, la similarité cosinusoïdienne, la distance de Manhattan, etc., puis regroupe les points avec le score de similarité le plus élevé.

Non supervisé - Clustering

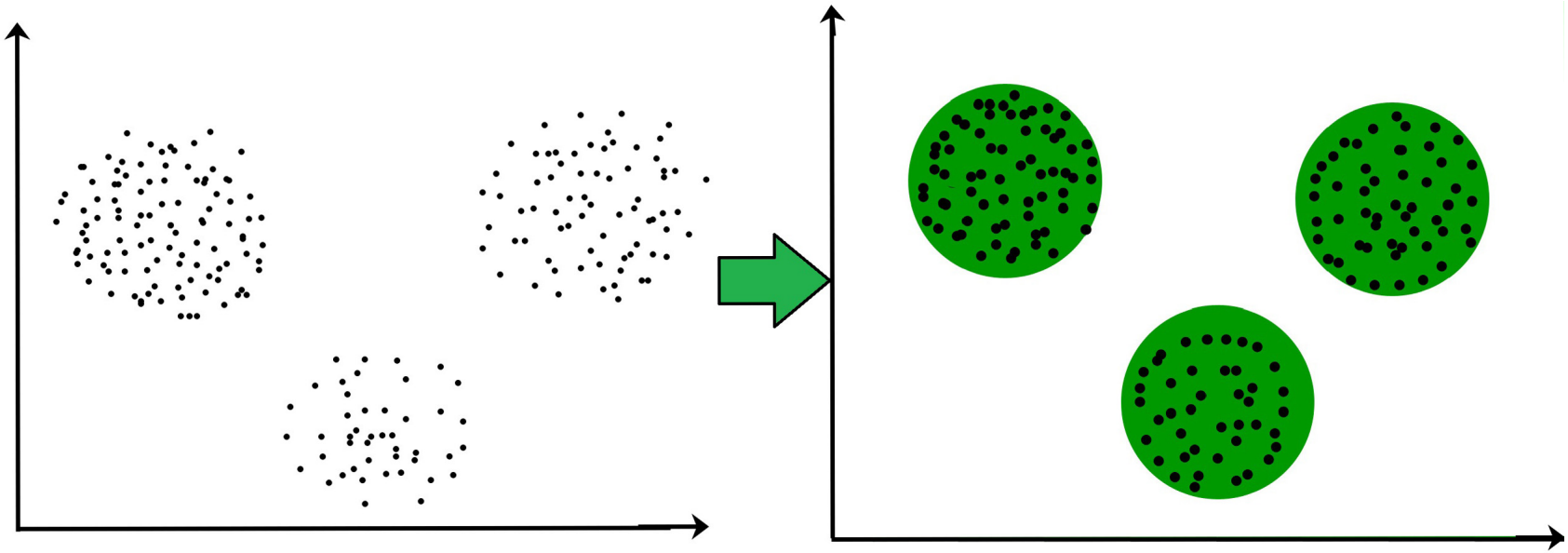

Figure: https://www.geeksforgeeks.org/clustering-in-machine-learning/

Non supervisé - Clustering

Types de clusterings :

- “Soft” : Propose une probabilité de faire partie d’un cluster.
- “Hard” : Traditionnel, assigné à X cluster.
- Hiérarchique : Par profondeur/”verticalité” pour consolider les clusters (division ou absorption).
- Etc.

Principal Component Analysis (PCA)

L'Analyse en composantes principales (réduction de la dimensionnalité) est une technique utilisée pour réduire le nombre de caractéristiques dans un ensemble de données tout en conservant autant d'informations importantes que possible.

En d'autres termes, il s'agit d'un processus de transformation de données de grande dimension en un espace de dimension inférieure qui préserve toujours l'essence des données d'origine.

Non supervisé - Principal Component Analysis (PCA)

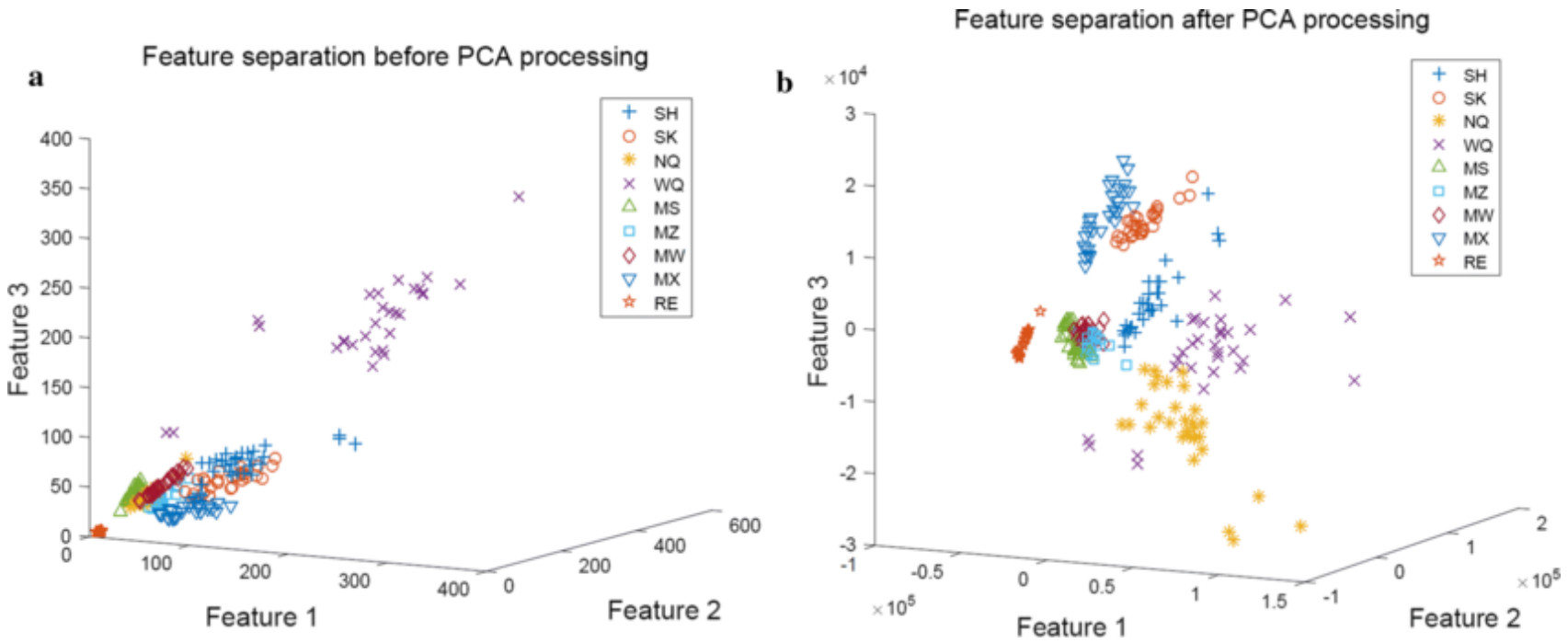

Figure: `https://www.researchgate.net/figure/Comparison-of-feature-separation-before-and-after-PCA-dimension-reduction_fig8_331794885`

Non supervisé - Principal Component Analysis (PCA)

PCA sur MNIST pour réduire de 8x8 = 64 dimensions (chaque pixel compte comme une dimension) à 2 dimensions

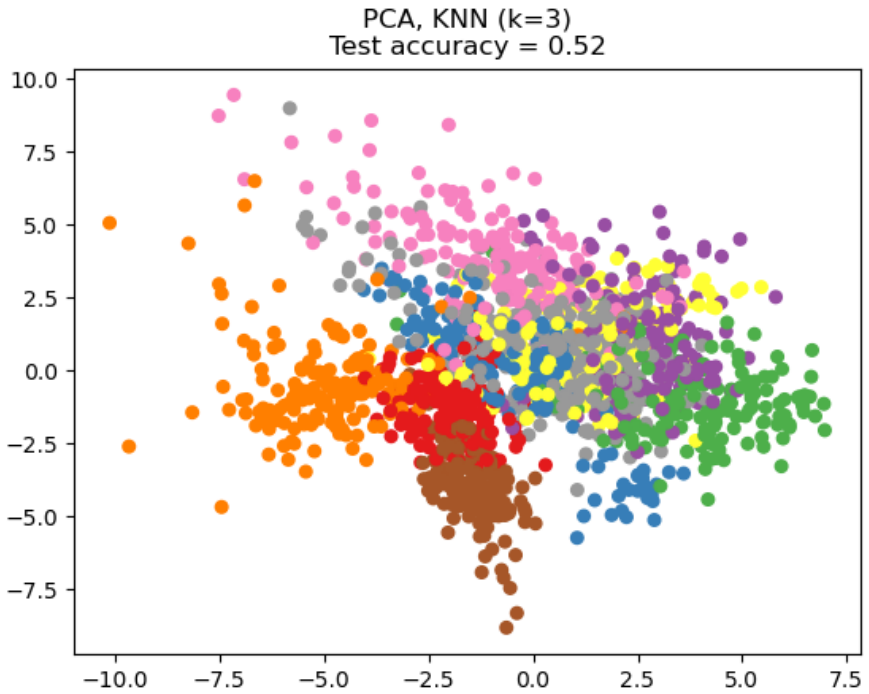

Figure: `https://scikit-learn.org/stable/auto_examples/neighbors/plot_nca_dim_reduction.html`

Détection d'Anomalies

Une méthode très intéressante qu'on oublie souvent, mais qui est très logique, est la détection d'anomalies.

Desfois, s'il est trop dur de trouver un lien, pourquoi ne pas faire... l'inverse!

Trouver le négatif de quelque chose (le défaut) est généralement plus facile.

Détection d'Anomalies - Principe

On considère avoir un "dataset" pur et contenant une distribution complètement de confiance.

Le principe fondamental serait que peu importe la méthode utilisée pour détecter un "outlier" de la distribution des données (distance, etc.), la valeur détectée ainsi est rejetée.

Détection d'Anomalies - Exemple

N'oubliez pas d'exploiter des astuces comme le changement de représentation et/ou dimensions.

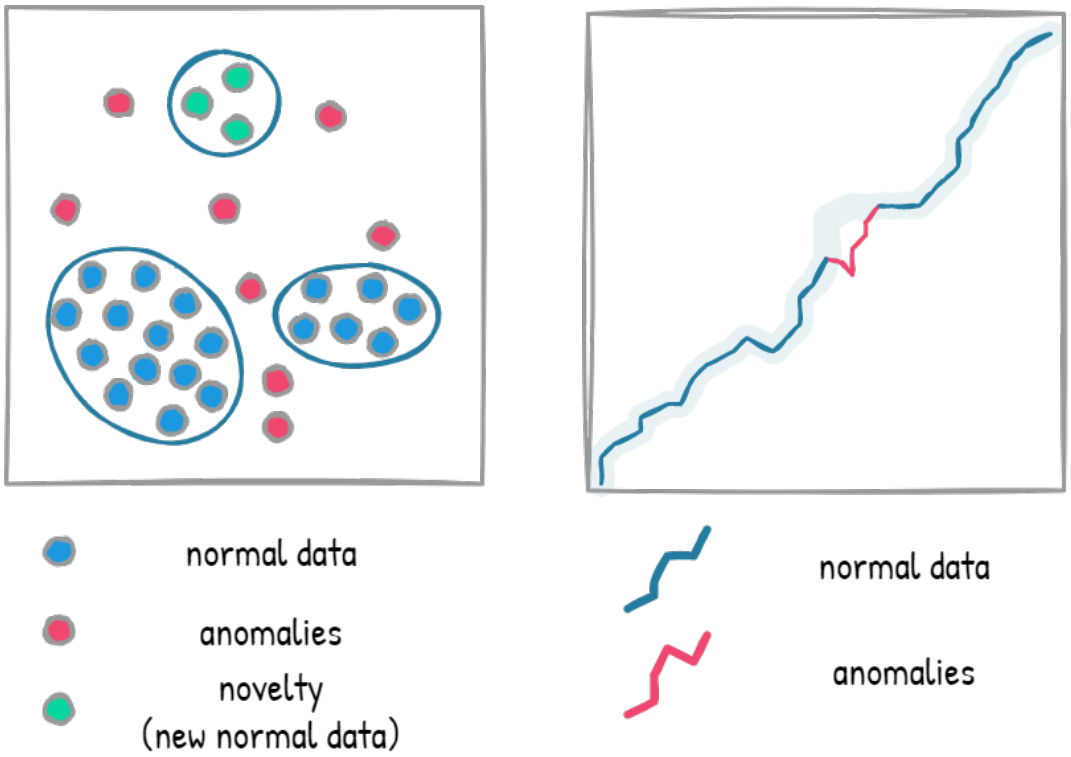

Figure: `https://mlguru.ai/Learn/ai-use-cases-anomaly-detection`

Algorithme “Apriori”

Comme nous avons mentionné auparavant, **les données sont très importantes** puisqu'elles déterminent si un algorithme va bien performer.

Rappelons-nous que le modèle ajuste ses “poids” grâce aux données qu'il injeste et retient lors de l'entraînement.

On peut le considérer comme du ML :
`https://www.ibm.com/think/topics/apriori-algorithm`

Algorithme “Apriori”

Utilité : Des situations avec le principe de transactions avec des fréquences d'items (peu importe la forme) et des possibilités multiples.

Exemple commun : Systèmes de recommendations pour les achats dans les commerces divers (épiceries, e-commerce, etc.).

Exemple reconnu : Achat de bière et couches et l'organisation spécifique des magasins de surfaces.

Algorithme “Apriori” - Exemple

Pour une liste de transactions dans notre site Web E-Commerce :

- Bureau, Clavier, Écran
- Bureau, Clavier
- Bureau, Écran
- Clavier, Écran
- Bureau, Écran

Algorithme “Apriori” - Exemple

On compte le nombre de transactions pour chaque item de la base de données :

Item	Compte	%
Bureau	4	80%
Clavier	3	60%
Écran	4	80%

Algorithme “Apriori” - Exemple

On peut le faire pour un ensemble de 2, 3, etc.

Voici un exemple pour 2 :

Items (Paires)	Compte	%
Bureau, Clavier	2	40%
Bureau, Écran	3	60%
Clavier, Écran	2	40%

Et pour 3 :

Items (Paires)	Compte	%
Bureau, Clavier, Écran	1	20%

Algorithme “Apriori” - Exemple

Finalement, on peut avoir deux (2) types de seuils que l'on peut choisir arbirairement :

- Support : pour les items et les pairs.
  - Seuil choisi : 50%
- Confidence : pour les suites d'événements “Apriori”.
  - Seuil choisi : 60%
- Lift : Puissance de l'association (écarte la possibilité que c'est simplement de la chance)
  - Ratio de 1 ou plus grand == association forte (attention, voir avec les autres règles s'il y a un nombre beaucoup plus grand)

Algorithme “Apriori” - Exemple

Les formules pour référence :

$$\text{Support}(I) = \frac{\text{Occurrences de l'item I dans la liste de transactions}}{\text{Nombre total de transactions}}$$

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Transaction avec les items de X et Y (sans ordre)}}{\text{Transaction avec les items de X}}$$

$$\text{Lift} = \frac{\text{Confidence}}{\text{Support}}$$

Algorithme “Apriori” - Exemple

Voici ce qu'on peut conclure :

- Support :
  - On ignore la paire de trois (3) items puisque $< 50\%$
- Confidence :
  - Règle 1 : Achat d'un Bureau $\rightarrow$ Clavier
    - Support (Bureau, Clavier) $==$ 2
    - Support (Bureau) $==$ 4
    - Confidence $= 2/4 = 50\%$ (**Échec**)
  - Règle 2 : Achat d'un Clavier $\rightarrow$ Bureau
    - Support (Bureau, Clavier) $==$ 2
    - Support (Clavier) $==$ 3
    - Confidence $= 2/3 = 66\%$ (**Réussite**)
- Lift :
  - Règle 1 : $0.5/0.4 = 1.25$
  - Règle 2 : $0.66/0.4 \approx 1.65$

Algorithme “Apriori” - Exemple

Conclusion : Acheter un Clavier avant offrirait plus de chances de sécuriser un achat d'un Bureau. Acheter le Bureau avant offre moins de chances puisque la base de données de transactions contient des paires Bureau/Écran venant réduire les chances.

Attention : Cet exemple est trop naïf et fictif pour inférer quoi que ce soit de plus logique. De plus, les données de transactions sont complètement fausses.

Algorithme “Apriori” - Exemple

Comment faire pour les paires de trois (3) items et plus ?

Il faut simplement les considérer comme un groupe puisque l'algorithme est itératif, et nous avons la réponse précédente.

Exemple fictif (Emplacement de la flèche dicte le dénominateur) :

- Confidence :
  - Règle 1 : Achat d'un Bureau $\rightarrow$ (Clavier, Écran)
    - Support (Bureau, Clavier, Écran) $==$ 1
    - Support (Bureau) $==$ 4
    - Confidence $= 1/4 = 25\%$ (**Échec**)
  - Règle 2 : Achat d'un (Bureau, Clavier) $\rightarrow$ Écran
    - Support (Bureau, Clavier, Écran) $==$ 1
    - Support (Bureau, Clavier) $==$ 2
    - Confidence $= 1/2 = 50\%$ (**Échec**)

Bibliographie

- https://www.geeksforgeeks.org/ml-classification-vs-regression/