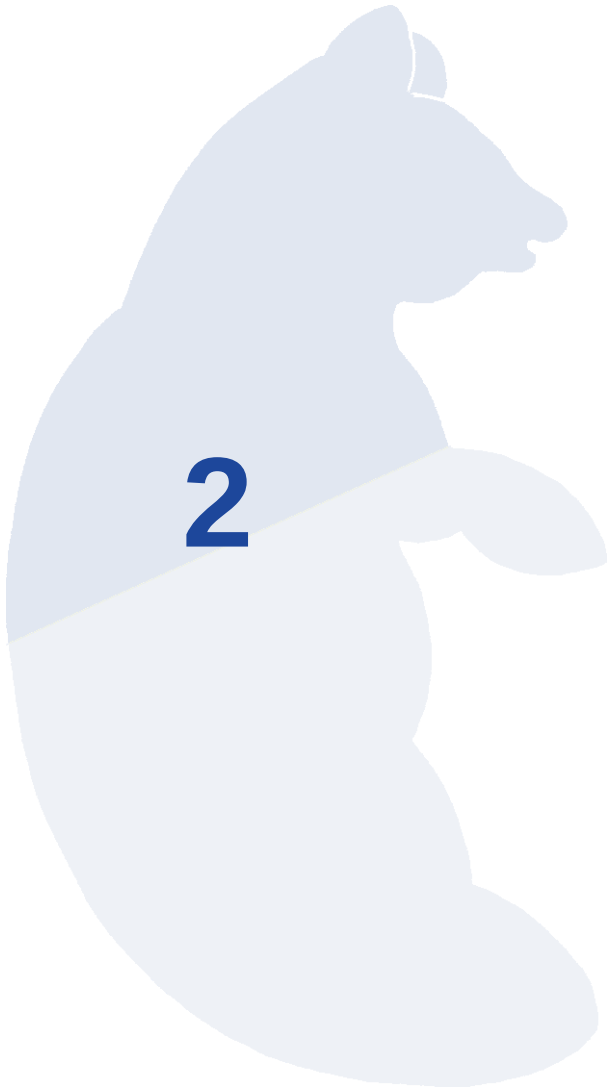


t-SNE

2. t-SNE

L’algorithme de réduction de dimensionnalité appelé **t-distributed stochastic neighbor embedding** (**t-SNE**) est un algorithme d’apprentissage non supervisé.

Développé par Laurens van der Maaten et Geoffrey Hinton, il permet d’analyser des données décrites dans des espaces à forte dimensionnalité (via un grand nombre de descripteurs) pour les représenter dans des espaces à deux ou trois dimensions.

Cet algorithme est très utilisé car il facilite la visualisation de données ayant beaucoup de descripteurs.

t-SNE est un **algorithme non-linéaire de “feature extraction”** qui construit une nouvelle représentation des données de telle sorte que les données proches dans l’espace original aient une probabilité élevée d’avoir des représentations proches dans le nouvel espace. A l’inverse, les données qui sont éloignées dans l’espace original, ont une probabilité faible d’avoir des représentations proches dans le nouvel espace.

En pratique la similarité entre chaque paire de données, dans les deux espaces, est mesurée par le biais de calculs probabilistes basés sur des hypothèses de distribution. Et les nouvelles représentations se construisent de telle sorte à minimiser1 la différence2 entre les distributions de probabilités mesurées dans l’espace original et celles du nouvel espace.

2. t-SNE

Cependant, même si cet algorithme crée une distribution qui respecte la proximité entre les objets les plus proches, la nouvelle représentation ne respecte pas forcément les distances et les densités de distribution des données originales.

Cette méthode est très lourde en termes de calcul, ce qui limite (sérieusement) l'utilisation de cette technique : dans le cas de données de très haute dimension, une autre technique de réduction de la dimensionnalité devra être pratiquée avant d'utiliser t-SNE.

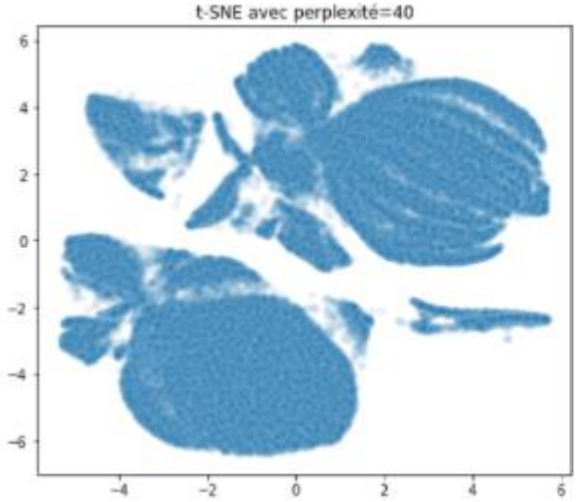
2. t-SNE

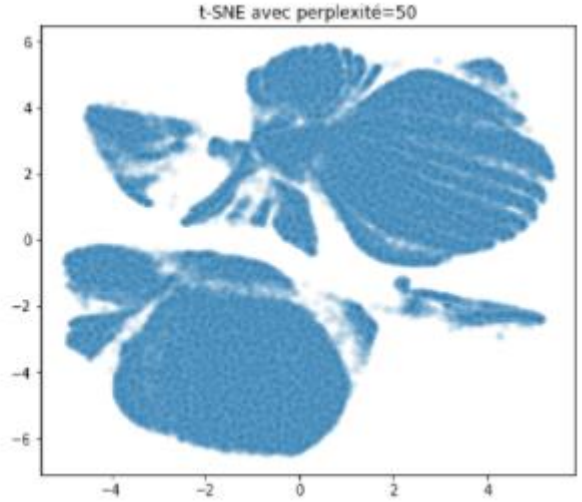
Hyper-paramètres :

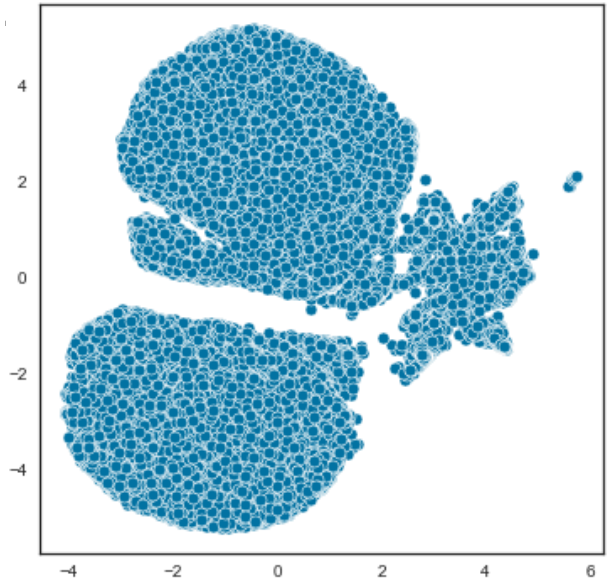
- **n_components** : dimension de l'espace incorporé (int, optionnel, défaut :2).
- **perplexity** : la perplexité est liée au nombre de voisins les plus proches qui est utilisé dans d'autres algorithmes d'apprentissage multidimensionnel.Envisagez de choisir une valeur entre 5 et 50. Le choix n'est pas extrêmement critique puisque t-SNE est assez peu sensible à ce paramètre (float, optionnel, défaut : 30).
- **early_exaggeration** : Contrôle l'étroitesse des clusters naturels de l'espace original dans l'espace incorporé et l'espace qui les sépare. Pour de plus grandes valeurs, l'espace entre les clusters naturels sera plus grand dans l'espace incorporé. Encore une fois, le choix de ce paramètre n'est pas très critique. Si la fonction de coût augmente pendant l'optimisation initiale, le facteur d'exagération précoce ou le taux d'apprentissage pourrait être trop élevé (float, optionnel, défaut: 4.0).
- **learning_rate** : le taux d'apprentissage peut être un paramètre critique. Il doit être compris entre 100 et 1000. Si la fonction de coût augmente pendant l'optimisation initiale, le facteur d'exagération précoce ou le taux d'apprentissage peut être trop élevé. Si la fonction de coût reste bloquée dans un mauvais minimum local, l'augmentation du taux d'apprentissage est parfois utile (float, optionnel, défaut: 1000).
- **n_iter** : nombre maximal d'itérations pour l'optimisation. Doit être d'au moins 200 (int, optionnel, défaut: 1000).

2. t-SNE

- **metric** : la métrique à utiliser pour calculer la distance entre les instances d'un tableau de caractéristiques (string or callable, optionnel, défaut: "euclidean")
- **init** : initialisation de l'incorporation. Les options possibles sont 'random' et 'pca'. L'initialisation PCA ne peut pas être utilisée avec des distances précalculées et est généralement plus stable globalement que l'initialisation aléatoire (string, optionnel (défaut: "random").
- **verbose** : niveau de verbosité (int, optionnel, défaut: 0).
- **random_state** : nombre aléatoire (int ou RandomState instance ou None (défaut)







5. t-SNE

5. t-SNE