EPITECH INNOVATIVE PROJECT

Rapport

MLII_Unsupervised_Learning_and_Agents

23 / 12 / 2023 Killian VALLETTE, Simon GUYADER

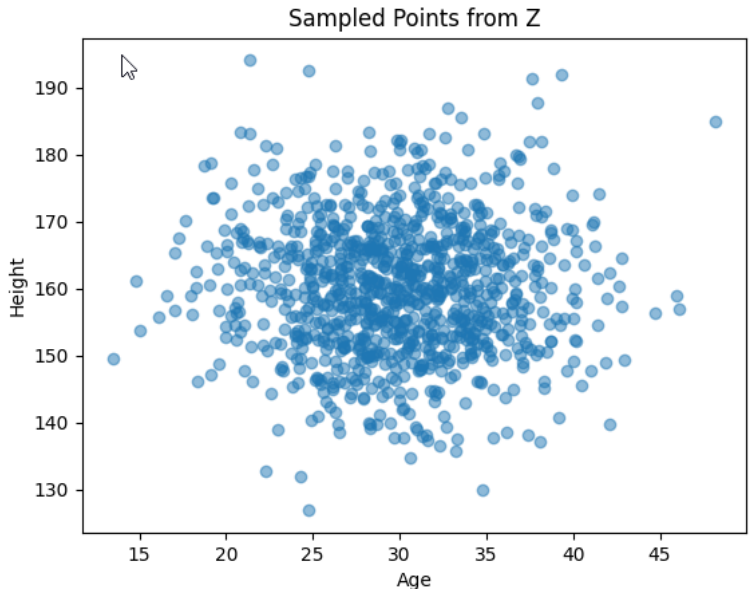
Exercice 1:

1. Distribution :

On débute avec une distribution bidimensionnelle, Z = (X, Y), où X et Y sont des variables aléatoires représentant par exemple l'âge et la taille d'une population. Le code génère 1000 points à partir de cette distribution en utilisant des nombres aléatoires.

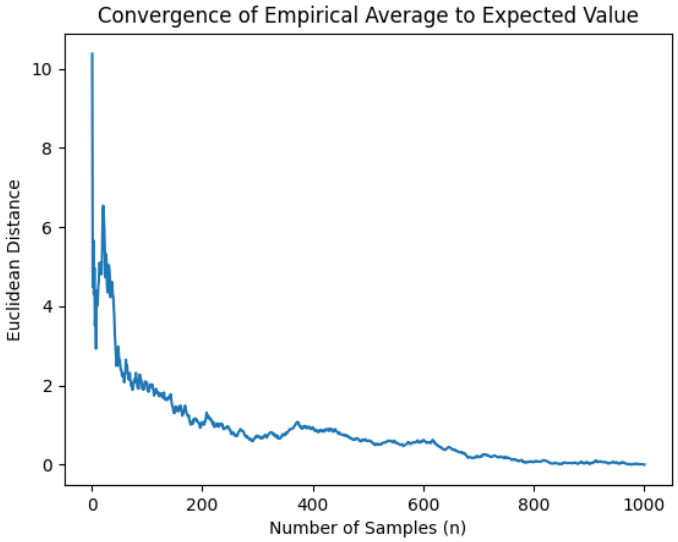
2. Espérance et Visualisation :

On calcule l'espérance de Z, qui est essentiellement le centre de gravité des variables. Les points générés sont ensuite représentés dans un graphique 2D, avec l'âge sur l'axe des x et la taille sur l'axe des y. Cela donne une représentation visuelle des données.



3. Convergence de la Moyenne Empirique :

On examine si la moyenne empirique (une moyenne calculée à partir des échantillons) converge vers l'espérance théorique à mesure que le nombre d'échantillons augmente. On mesure la proximité en calculant des distances euclidiennes entre la moyenne empirique et l'espérance pour différentes tailles d'échantillons, puis on représente graphiquement cette convergence.



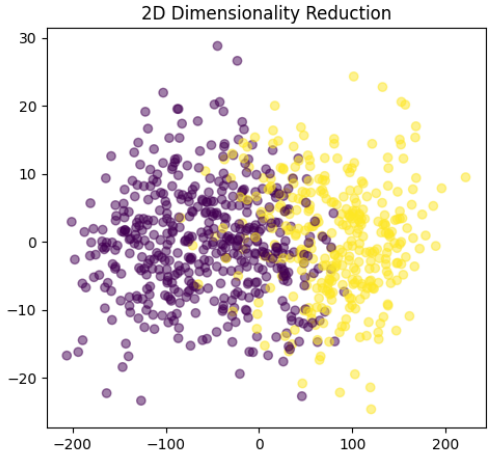
Conclusion : En résumé, l'exercice démontre comment, en accumulant davantage de données, notre estimation tend à se rapprocher de la réalité. C'est un peu comme rendre une photo floue plus nette en ajoutant des photos similaires. C'est la magie de la loi des grands nombres à l'œuvre !

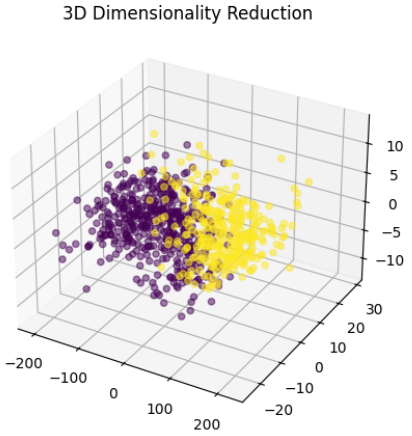
Exercice 2:

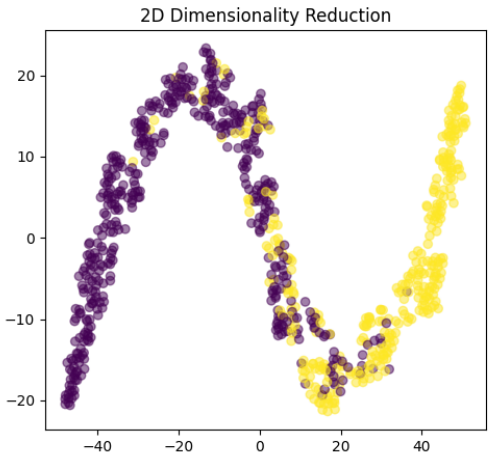
Méthodes de Réduction de Dimensionnalité :

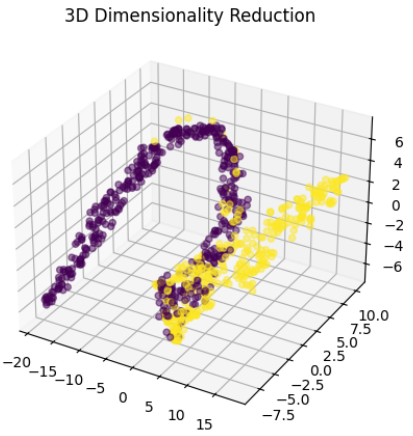
Nous utilisons deux méthodes : l'Analyse en Composantes Principales (PCA) et le t-Distributed Stochastic Neighbor Embedding (t-SNE).

1. PCA :
    - On utilise la PCA pour réduire les données à 2 et 3 dimensions.
    - Les résultats sont visualisés à l'aide de graphiques de dispersion en 2D et 3D, où les points sont colorés en fonction des étiquettes d'origine (présence ou absence de tempête).
2. t-SNE :
    - Ensuite le t-SNE pour une autre réduction de dimension à 2 et 3 dimensions.
    - Les résultats sont visualisés de la même manière.









Visualisation :

- Les graphiques de dispersion 2D montrent une séparation plus claire des classes avec PCA par rapport à t-SNE.
- Cependant, dans les graphiques 3D, la séparation semble plus nette avec t-SNE.
- La dimension 3 à l'air de donner une meilleure séparation des classes.

Conclusion :

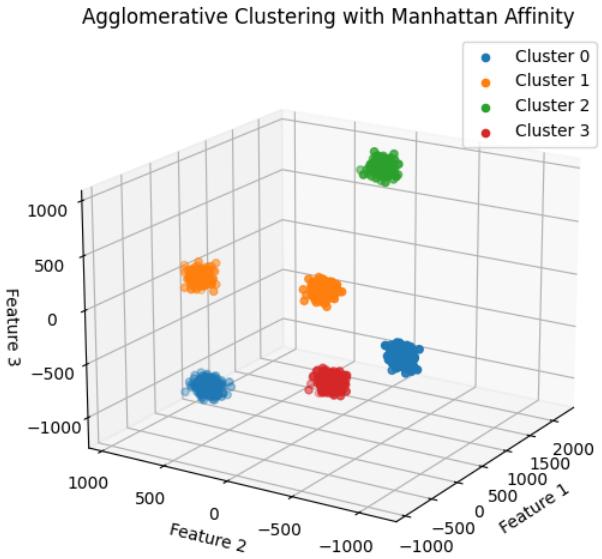
En conclusion, la dimension 3, obtenue à partir de t-SNE, semble plus prometteuse pour la prédiction des tempêtes basée sur les composants projetés. Bien que PCA fournisse une séparation claire en 2D, la dimension supplémentaire offerte par t-SNE semble apporter des informations supplémentaires bénéfiques pour la prédiction. Les résultats de PCA et t-SNE, bien que différents, offrent des perspectives intéressantes pour la réduction de dimensionnalité dans ce contexte.

Exercice 3:

Objectif:

L'objectif de l'exercice est de discuter de différentes méthodes pour déterminer le nombre optimal de clusters d'un dataset.

Voici la représentation du jeu de données:



Il fallait choisir 2 méthodes de clustering et 2 méthodes heuristiques pour déterminer le nombre de clusters.

Les 2 méthodes de clustering que nous avons choisis sont :

- Kmeans
- Agglomerative Hierarchical

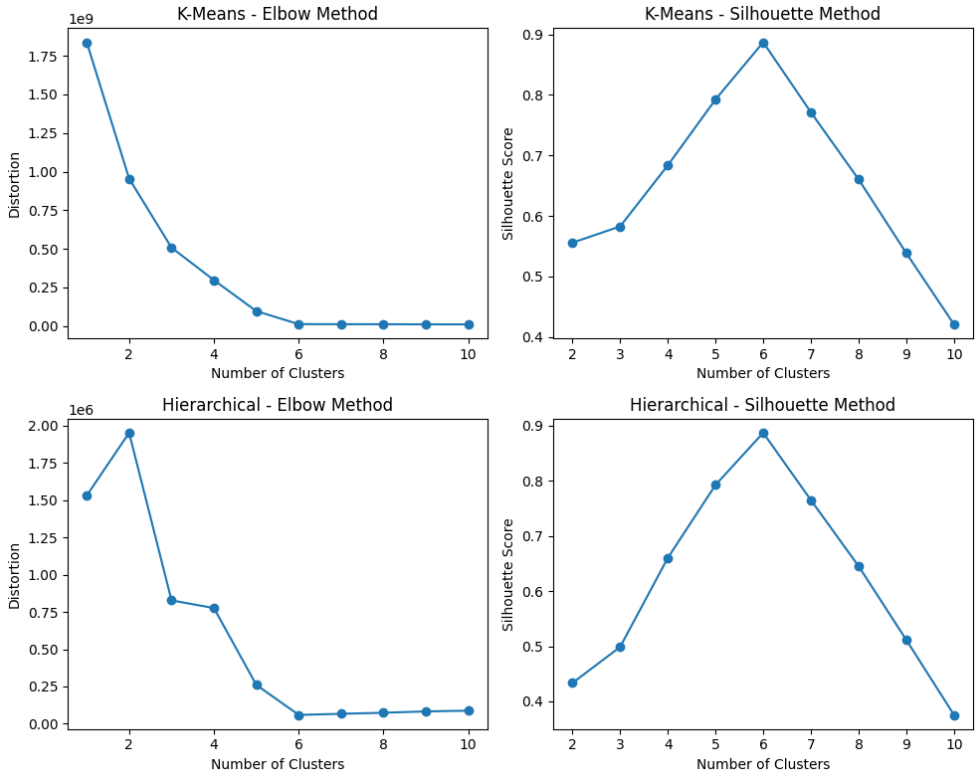
Les 2 méthodes heuristiques sont :

- La méthode “Elbow”
- La méthode “Silhouette”

Les metrics utilisés diffèrent entre Kmeans et Agglomerative Hierarchical,

- Kmeans : Euclidien
- Agglomerative Hierarchical : Manhattan

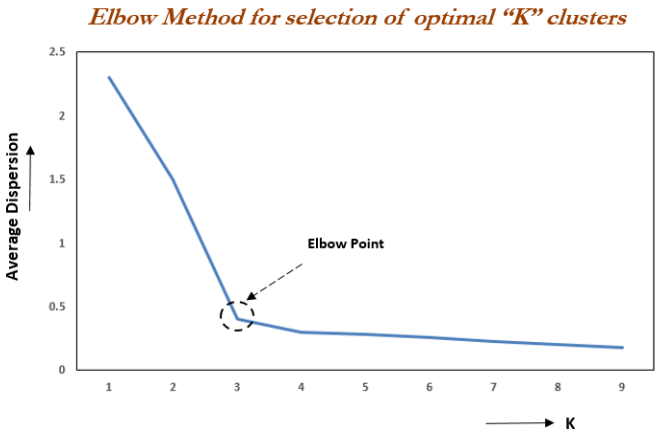
Les résultats obtenus:



Interprétation des résultats:

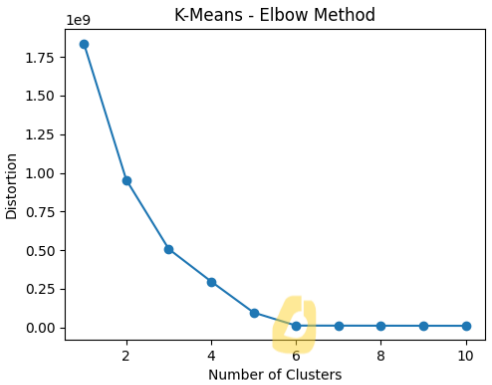
Elbow method:

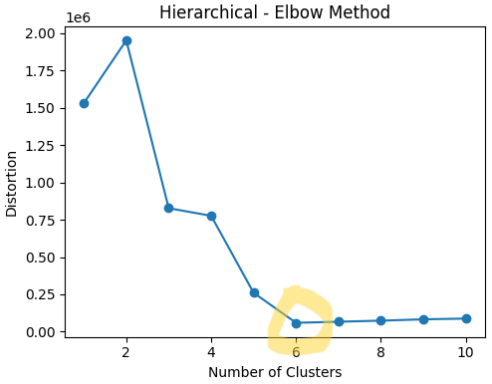
L'objectif de la méthode est de trouver le point de cassure de la courbe pour obtenir le nombre optimal de cluster



Pour notre exemple :

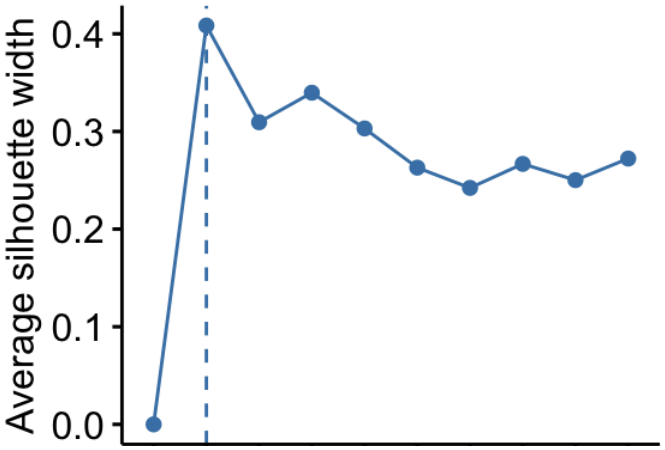
Nous voyons que pour l'algorithme K Means et AH (Agglomerative Hierarchical), le point "elbow" se situe à 6 de cluster, cependant nous observons une différence de courbe qui est plus lisse du côté du Kmeans, sans doute expliqué par la nature et le metric de l'algorithme.





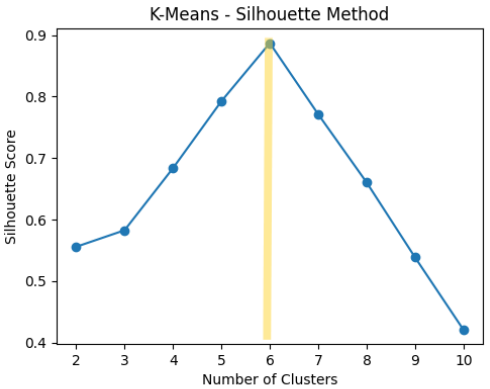
Silhouette method:

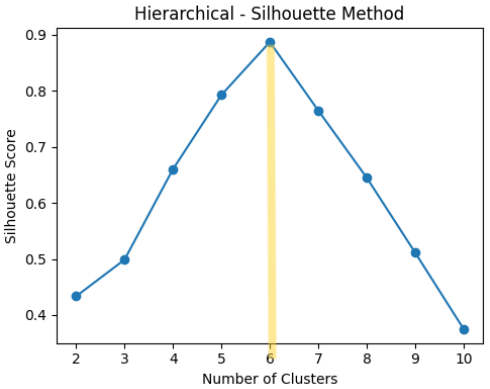
L'objectif de cette méthode est de mesurer la qualité de partition des données, cette mesure renvoie un coefficient compris entre -1 (pire classification) et 1 (meilleure classification).



Pour notre exemple :

Nous voyons que pour l'algorithme K Means et AH (Agglomerative Hierarchical), le meilleur score se situe à 6 de cluster aussi car le score est le plus proche de 1





Conclusion:

La mise en comparaison de ces différentes méthodes semblent montrer que la combinaison de l'algorithme K Means et l'heuristique Silhouette offre la représentation la plus clair et lisible

Exercice 4:

Description:

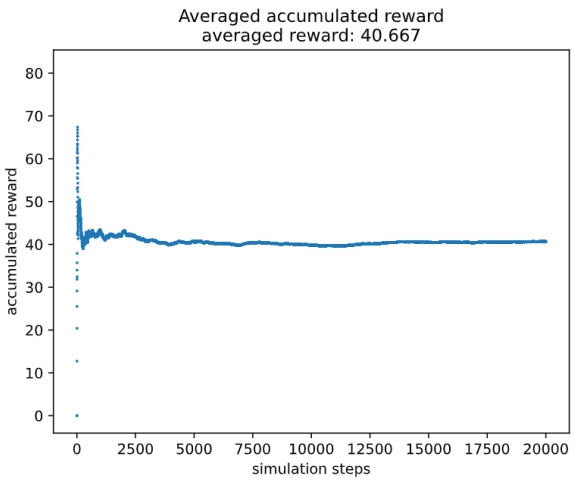
- Monde en 1 dimension de taille 8
- On gère un Agent
- Déplacement: gauche, droite ou pas de déplacement
- Des récompenses sur certaines cases
- En arrivant sur une récompense on la mémorise
- Les récompenses changent aléatoirement de temps en temps

Objectif:

- Obtenir un score final moyen d'au moins 20.

Déroulé:

- On test quelques algo pour comprendre comment fonctionne le score
  - déplacement uniquement droite
  - déplacement uniquement gauche
  - déplacement aléatoire
  - etc…
- On arrive pas a atteindre plus de 16.5
- On essaye de comprendre le fichier *simulation.py*
- On remarque ligne 60 et 66 que si on finit un step sur une récompense on augmente le score moyen
- On change l'algo pour que quand on arrive sur une récompense on bouge plus
- On passe à >40 de score



Exercice 5, Cas d’étude: