

## Explication globale du fichier modele.ipynb

Le document **model.ipynb** contient l'ensemble du code utilisé dans ce projet. Le dossier **data** renferme tous les fichiers CSV nécessaires, y compris le fichier **data\_GPU.ipynb**, qui permet de modifier et de gérer la table d'efficacité du matériel des GPU. Ce dernier crée le fichier **data\_GPU.csv** dans le dossier **csv\_files**. Il inclut également des données pour analyser la consommation des GPU, comme les émissions de CO2 lors de la production, la consommation d'eau, l'épuisement des ressources abiotiques, et les déchets électroniques produits. Chaque fichier de données est stocké avec un nom équivalent en anglais dans le dossier **csv\_files**, situé dans le même répertoire que **data**.

Le code dans **model.ipynb** est structuré en plusieurs catégories :

- Données et fonctions : Cette section contient le code pour lire les fichiers CSV et les fonctions permettant de traiter les données. Ces fonctions sont regroupées en sous-catégories selon leur utilisation.
- Modèle : Dans cette partie, un DataFrame est créé pour chaque année, regroupant les données liées au datacenter. Les algorithmes utilisent les fonctions déclarées dans la section précédente.
- Visualisation des données : Cette section permet de visualiser les résultats sous forme graphique et d'interagir avec différents paramètres de manière interactive. Toutes les fonctions nécessaires à la création des graphiques sont définies ici.
- Fonction de calcul de la surface en m<sup>2</sup> utilisée par le datacenter : Cette partie modélise l'organisation de la salle contenant les serveurs et GPU (on suppose que tous les GPU se trouvent dans une seule salle). Toutes les fonctions nécessaires à la création des graphiques relatifs à l'espace occupé sont définies dans cette section.

Dans la partie Modèle, la fonction **model** renvoie plusieurs éléments, dont un DataFrame contenant les données pour chaque mois concerné. La fonction **create\_df\_test\_model** utilise cette première fonction pour ajouter une ligne au DataFrame pour chaque mois écoulé, avec les données correspondantes. Cela permet de stocker une synthèse complète dans le DataFrame, couvrant toute la période de l'expérience, avec chaque ligne correspondant à un mois précis.

Tout au long du code, les variables statiques sont stockées dans trois dictionnaires principaux :

- **dict\_variables** : Situé dans la partie Données et fonctions, il contient la majorité des variables statiques utilisées dans cette section ainsi que dans la partie Modèle.
- **dict\_indicators** : Situé également dans la partie Données et fonctions, ce dictionnaire stocke le nom du fichier contenant les données relatives à chaque indicateur qui mesure la consommation du datacenter.
- **dict\_variables\_servers** : Situé dans la partie Fonction de calcul de la surface en m<sup>2</sup> utilisée par le datacenter, ce dictionnaire contient toutes les variables statiques en lien avec l'organisation et la surface occupée par les racks de serveurs dans le datacenter.

Les valeurs de ces variables doivent être modifiées directement lors de la déclaration des dictionnaires pour faciliter l'organisation du code.