**Classification automatique de la thermosensibilité de la consommation électrique de bâtiments.**

Le but de ce TD est de vous familiariser avec l’utilisation d’un algorithme de clustering (KMeans) avec pour application *la classification automatique de la thermosensibilité de consommation de bâtiments[[1]](#footnote-0)*: on se propose d'entraîner un algorithme à automatiquement reconnaître si un bâtiment est chauffé à l’électricité, si de la climatisation est utilisée, si le chauffage est un usage principal… S' il vous reste du temps vous pourrez aussi vous intéresser, avec le même genre de techniques, à la classification de la *régulation de la consommation*.

1. Ouvrez le dataset deepki\_diderot\_daily\_data.csv et explorez les données.
2. A partir des données, obtenez pour chaque bâtiment le profil de consommation du bâtiment en fonction de la température extérieure (*signature énergétique*) en utilisant un échantillonnage de la température avec un . Plottez quelques exemples de signature énergétique.
3. Pour réaliser le clustering, il est pertinent que les consommations des bâtiments soient obtenues pour une même plage de température fixée (par ex avec un échantillonnage ). Quelle technique pouvez-vous utiliser pour ça ? Transformez les données de cette manière.
4. Après avoir rescalé (centré et réduit) les profils de consommation de chaque bâtiment, réalisez un clustering en utilisant KMeans avec k = 3.
5. Plottez les signatures énergétiques des bâtiments dans différents clusters, comparez aussi les profils de consommation moyens (centre de chaque cluster). Comment interpréter la classification ? Faites varier le nombre de cluster k et regardez comment les résultats évoluent. Trouvez une ou des estimation(s) du nombre optimal théorique de clusters en utilisant KElbowVisualizer (cherchez un exemple sur internet. Plusieurs métriques sont possibles et donneront potentiellement des résultats différents).
6. Définissez deux features x1 et x2 que vous utiliserez pour construire une représentation 2D (scatter plot) des bâtiments selon leurs valeurs de x1 et x2. Représentez leur appartenance aux différents clusters par des groupes de couleurs. Choisissez un couple de features x1 et x2 pertinent qui permette de visualiser dans une certaine mesure la ségrégation des clusters.
7. Ici le clustering a permis de classifier l’évolution en fonction de la température de la consommation des bâtiments. Comparez les consommations surfaciques moyennes des bâtiments dans chaque cluster : la thermosensibilité a-t-elle une influence sur le niveau absolu de consommation ?
8. En option, s'il vous reste du temps : Vous pouvez aussi explorer le jeu de données deepki\_diderot\_hourly\_data.csv qui contient des consommations au pas de temps horaire pour faire une classification du profil de consommation journalier. Vous pouvez par exemple filtrer les données des premières 24 heures du jeu de données et les analyser.
9. Interprétez les clusters identifiés. Plottez les profils de consommation horaire des bâtiments dans différents clusters, comparez aussi les profils de consommation moyens (centre de chaque cluster)
10. Comme précédemment, définissez deux features x1 et x2 que vous utiliserez pour construire une représentation 2D (scatter plot) des bâtiments selon leurs valeurs de x1 et x2. Représentez leur appartenance aux différents clusters par des groupes de couleurs. Choisissez un couple de features x1 et x2 pertinent qui permette de visualiser dans une certaine mesure la ségrégation des clusters.

**Quelques fonctions et packages clés :**

* Pour manipuler et transformer les données : pandas et numpy ; les méthodes mean() et groupby() et interpolate() de pandas notamment
* Visualisations : matplotlib.pyplot et/ou seaborn
* Clustering : sklearn.cluster.KMeans ; pour estimer le nombre optimal de cluster la méthode KElbowVisualizer du package yellowbrick.cluster peut être utilisée.

1. Pour aller plus loin : Westermann et al, *Unsupervised learning of energy signatures to identify the heating system and building type using smart meter data*, Applied Energy 264 (2020) 114715. [↑](#footnote-ref-0)