# Explication des codes du projet Galeo

## Fichiers du GitHub

Les fichiers du GitHub suivent l’ordre suivant :

1. Build\_one\_data\_set
2. Pipeline Data Preparation
3. Exploratory Data Analysis
4. Get Excel from Exploratory Data Analysis
5. Machine Learning

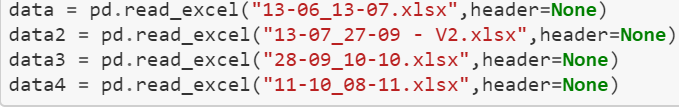
Chaque fichier est commenté et suit l’ordre suivant :

* Importation des librairies nécessaires
* Définition de chaque fonction utilisée
* Chargement de données (d’un Excel ou d’un précédent code au format « .p »)
* Lancement des fonctions
* Exportation des données obtenues au format « .p » ou Excel

1. Build\_one\_data\_set

**But** : Réunir les données des fichiers Excel dans une seule trame de données

**Données en entrées :** 4 fichiers Excel de juin à novembre



**Fonctions utilisées dans l’ordre du code :**

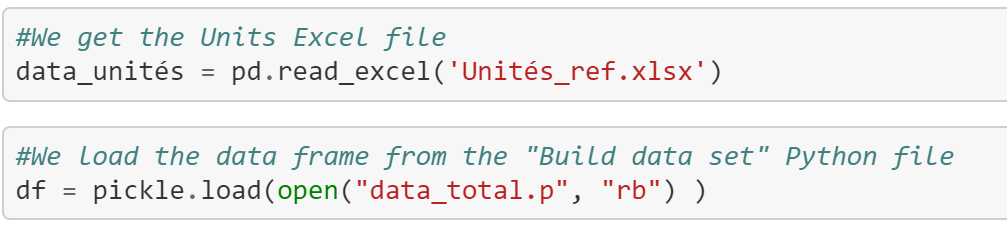
* « prepare\_data\_frame1 » : Réunion des deux premiers fichiers excels en une trame de données
* « prepare\_data\_frame2 » : Réunion des deux fichiers excels suivant les deux premiers en une trame de données
* « build\_one\_data\_set » : Construction d’une seule trame de données

**Données en sorties :** Une trame de données « data\_total » avec tous les fichiers réunis (sauvegardée dans « data\_total.p », une sauvegarde des noms des colonnes au cas où dans « columns\_names\_data\_total.p »

1. Pipeline Data Preparation

**But** : Préparer la trame de données pour l’analyse exploratoire

**Données en entrées :** Le fichier servant de référence pour les unités des adresses BACnet « Unités\_ref.xlsx » et la trame de données « data\_total.p » exportée du précédent code



**Fonctions utilisées :**

* « launch\_pipeline » : Lancement de toute la pipe line pour préparer les données c’est-à-dire :
  + Changement du format de la date en une seule colonne
  + Remplissage des valeurs nulles de façon linéaire
  + Ajout des adresses, noms en français et unités dans la trame de données
  + Suppression des colonnes dupliquées
  + Préparation de la trame de données en fichier Excel pour Power BI et simples observations dans un autre fichier Excel avec les adresses séparées en différentes colonnes, l’emplacement du point sur le plan, les unités, le mininum, le median et le maximum pour chaque point
  + Ajout des features « jour », « mois » et « saision » et récupération de toutes les énergies actives pour préparer la target
* « get\_target\_Energie\_totale » : Prépare la colonne de la valeur à prédire : l’énergie totale en faisant la somme des énergies actives

**Données en sorties :** La trame de données préparée « data\_total\_prepared.p » et la valeur à prédire, l’énergie dans « data\_total\_energie.p »

1. Exploratory Data Analysis

**But** : Faire de la réduction du nombre de features

**Données en entrées :** La trame de données « data\_total\_prepared.p » exportée du précédent code



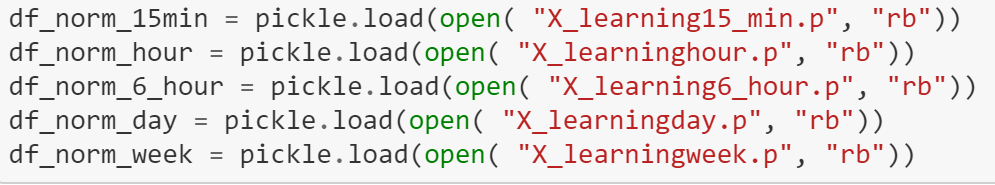
**Fonctions principales utilisées :**

* « prepare\_df\_to\_get\_correlations » : Préparer la trame de donnée avant de lancer l’analyse exploratoire (suppression de la colonne des dates, on garde en mémoire les adresses BACnet et les noms en français des points, Energie transformée en kw)
* « launch\_EDA» : Lancement des fonctions pour faire l’analyse exploratoire
  + Ajout de pas dans le temps : une heure, 6h, un jour et une semaine. Puis, normalisation.
  + on fait la matrice des corrélations entre les features et la target. On garde les features qui ont un seuil minimum de corrélation avec la target, ici l’énergie (0,7 pour tous les pas sauf pour le pas d’une semaine c’est 0,8). Il reste alors une centaine de features. On les met deux par deux en trouvant pour chaque feature à quelle autre feature, elle est le plus corrélée. Une fois que c’est fait, on garde seulement parmi ces features celle qui est le plus corrélée à la target.
* **Données en sorties :** Les data sets pour faire du Machine Learning avec les pas de : 15 min, 1 heure, 6 heures, 1 jour et 1 semaine

1. Get Excel from Exploratory Data Analysis

**But** : Extraire les features obtenues en fichier Excel

**Données en entrées :** Les 5 data sets obtenus du code précédent



**Fonctions utilisées :**

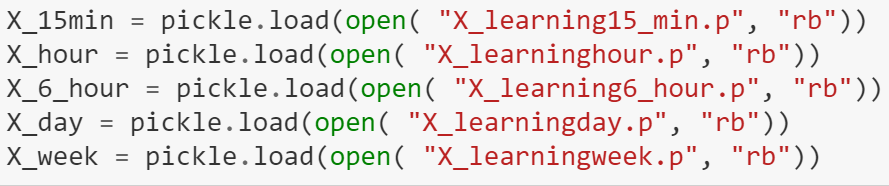
* « save\_df\_in\_excel » : Après avoir récupéré les noms des colonnes des 5 data sets, on les sauvegardes dans un Excel

**Données en sorties :** Le fichier « features.xlsx »

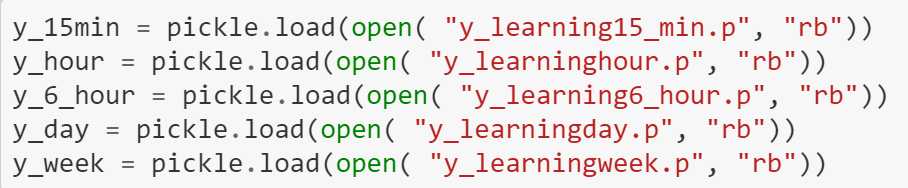
1. Machine Learning

**But** : Réaliser le modèle prédictif avec des algorithmes de Machine Learning

**Données en entrées :** Les 5 data sets obtenus du code précédent



Et leurs 5 valeurs à prédire également :



**Fonctions utilisées :**

* « SVR\_ML» : Réalisation d’un modèle de SVR (Support Vector Regression) avec en sortie le modèle obtenu, les résultats et un graphique des prédictions obtenues par rapport aux prédictions attendues
* « Linear\_Regression» : Même chose mais pour une Régression Linéaire
* « Decision\_Tree\_Regression » : Même chose mais pour un Arbre de décisions qui fait une régression

**Données en sorties :** Les modèle prédictifs avec l’algorithme le plus performant à choisir entre la SVR, la régression linéaire et le DTR (Decision Tree Regression) pour les pas de 15 min, 1 heure, 6 heures, 1 jour et 1 semaine