PRESENTATION DU PROJET DE PREDICTION DE CONSOMMATION

KARY-POTES Loris

Etudiant en alternance

en 1ère année d'école d'ingénieur

au Conservatoire National des Arts et Métiers

Ce document est une présentation d'un projet que j'ai réalisé en entreprise. J'ai choisi de présenter ce projet car c'est le plus concret et abouti que j'ai fait jusqu'ici. Je l'ai réalisé à l'aide de compétences que j'ai acquises en autodidacte. L'informatique et l'intelligence artificielle sont des domaines qui me passionnent, je passe une grosse partie de mon temps libre à apprendre et à me former sur ce sujet,

Sommaire

Introduction :

Dans l'entreprise d'accueil pour ma formation d'ingénieur j'ai été à l'initiative d'un projet de prédiction de consommation de vapeur industrielle pour un bâtiment de production d'insuline.

Le site de Fegersheim est spécialisé dans la production aseptique de médicaments injectables.

Je suis dans le service des énergies, rattaché à la direction technique du site.

Le département de l'énergie gère à la fois l'approvisionnement, la production et la distribution de diverses sources d'énergie nécessaire aux activités de l'entreprise et aux conditions de production des médicaments. La vapeur sert à chauffer les bâtiments et à humidifier les salles. Le chauffage s'effectue grâce à des centrales de traitement d'air, ce sont des caissons qui aspirent l'air extérieur, le réchauffent à l'aide d'un échangeur et le soufflent dans le bâtiment. L'échangeur est chauffé par la vapeur industrielle.

En tant qu'alternant dans ce service, nous avons comme mission de surveiller les consommations des bâtiments en diverses énergies. L'objectif de mon outil est de faciliter l'analyse et de faire remonter une consommation anormale plus rapidement. Le projet de base consistait à effectuer des prédictions et de les comparer avec les consommations réelles. Cette comparaison nous permettra de déterminer si une investigation doit être menée.

Données

Nous avons un outil en interne qui permet de récupérer les données des différents capteurs. Ces données sont stockées dans un serveur et nous y accédons depuis un logiciel appelé PI.

Les données qui ont une influence sur la consommation de vapeur du bâtiment sont la température extérieure, l'humidité extérieure et le pourcentage d'ouverture des vannes vapeurs sur chacune des centrales de traitement de l'air.

Pour relever ces données j'ai choisi de relever la valeur de chaque capteur toutes les 0.5 heures du 01/01/2018 au 01/01/2023.

Nous avons fréquemment des pertes de liaisons et des soucis liés au serveur. L'outil de relève remplace à ce moment les valeurs des capteurs pas des chaines de caractères comme :

« Arc Off-line », « No Data », « Comm Fail ».

J'ai donc appliqué un masque booléen pour remplacer chacune de ces valeurs par NA.

Voici un graphique qui représente la répartition des valeurs manquantes dans le dataset :

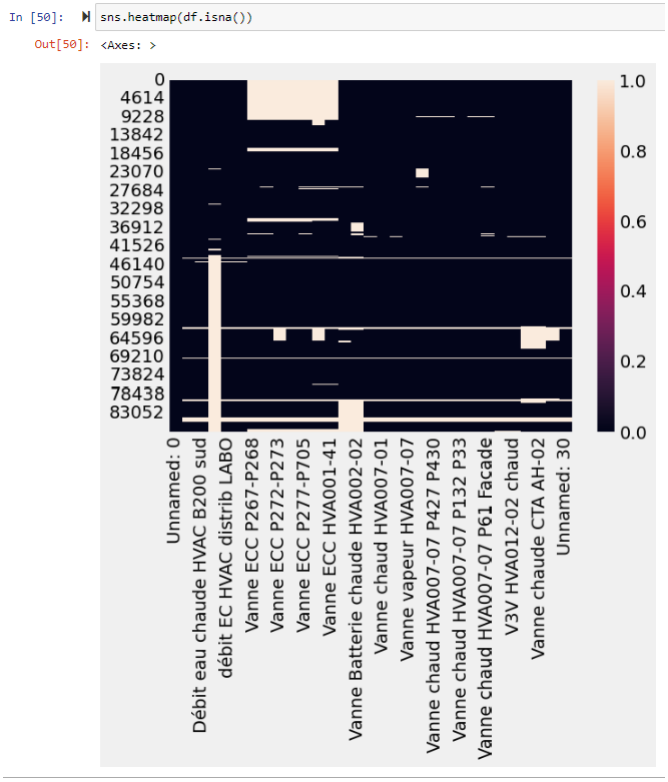

FIGURE 1: CARTE DE CHALEUR DES VALEURS MANQUANTES

Les valeurs manquantes étant regroupées, j'ai choisi de les supprimer.

Voici, en pourcentage, le taux de valeurs manquantes par variable.

```
In [49]: (df.isna().sum()/df.shape[0]).sort_values(ascending=True)

Out[49]: Unnamed: 0                          0.000000
         Unnamed: 30                         0.026561
         Débit B200                          0.026675
         Vanne chaud HVA007-07 P132 P33      0.027154
         Vanne chaud HVA007-07 P427 P430     0.027428
         Vanne chaud HVA007-07 P61 centre    0.027439
         Vanne chaud HVA007-07 P652 P653     0.027451
         Débit eau chaude HVAC B200 sud      0.027725
         débit EC HVAC distrib LABO          0.027747
         Vanne vapeur HVA007-07              0.027782
         débit EC labo chimie B200           0.027850
         Vanne chaud HVA007-01               0.028078
         Vanne chaud HVA007-01.1             0.028911
         Vanne vapeur HVA007-01              0.029231
         V3V HVA012-02 post chauffe          0.029904
         V3V HVA012-02 chaud                 0.030805
         Vanne chaud HVA007-07 P61 Facade    0.036224
         Vanne chaud HVA007-07 CTA           0.054399
         Unnamed: 29                         0.061370
         Vanne vapeur HVA037-00              0.097150
         Vanne chaude CTA AH-02              0.097321
         Vanne vapeur HVA002-02              0.115656
         Vanne Batterie chaude HVA002-02     0.140380
         Vanne ECC P275-P710                 0.170626
         Vanne ECC HVA001-41                 0.173763
         Vanne ECC P267-P268                 0.174277
         Vanne ECC P269                      0.175737
         Vanne ECC P277-P705                 0.177642
         Vanne ECC P272-P273                 0.206713
         Vanne vapeur HUM1 HVA001-41         0.226862
         débit EC HVAC B205                  0.513965
         dtype: float64
```

FIGURE 2: TAUX DE VALEURS MANQUANTES PAR VARIABLE

Pour conserver 80% du dataset de base, j'ai choisi de supprimer les colonnes avec plus de 20% de valeurs manquantes. J'ai ensuite supprimé toutes les lignes contenant les valeurs NA.

En plaçant la date en index, nous pouvons visualiser la cible.

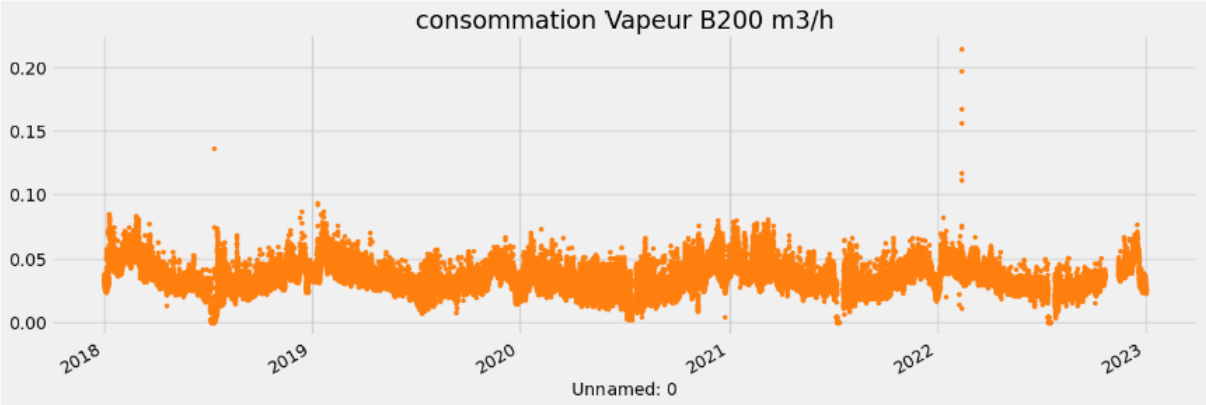

FIGURE 3: VISUALISATION DE LA CONSOMMATION VAPEUR EN M3/H DE 2018 A 2023

On s'aperçoit assez rapidement, visuellement, que nous avons des outliers, que j'ai choisi de trier manuellement en supprimant les valeurs supérieures à 0,085.

Ce graphique permet également d'apercevoir des phénomènes de saisonnalité.

Afin de pouvoir évaluer le modèle, je sépare le dataset en deux parties : une partie d'entrainement et une partie de test. Nous avons environ 5 années de données, en prenant une année pour le set de test cela équivaut à 20% de données servant à tester le modèle.

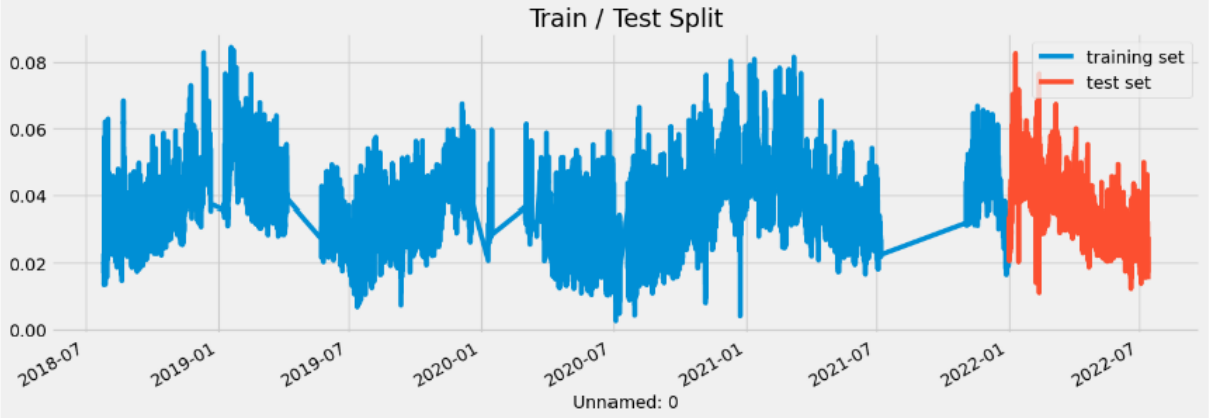

FIGURE 4: VISUALISATION DU SET D'ENTRAINEMENT ET DE TEST

Ajout de variables :

Pour améliorer la performance du modèle, j'ai rajouté des variables temporelles. En effet, les comportements de consommation d'énergie varient en fonction du temps, des cycles saisonniers et des jours de la semaine, et il est important de prendre en compte ces variations.

Par exemple, la consommation de vapeur industrielle peut varier en fonction de la journée de la semaine, avec des consommations plus élevées pendant les jours de semaine ouvrables par rapport aux week-ends. De même, la consommation peut varier en fonction du mois de l'année, avec des consommations plus élevées en hiver qu'en été.

Pour capturer ces variations, j'ai choisi de rajouter des colonnes pour chaque saisonnalité. Par exemple, la colonne 'dayofweek', représentant les jours de la semaine, peut prendre 7 valeurs, de 0 à 6 pour représenter chaque jour de la semaine. Les features rajoutées sont les suivantes :

```
In [108]:
def create_features(df):
    #create time series index
    df['hour'] =  df.index.hour
    df['dayofweek'] = df.index.dayofweek
    df['quarter'] = df.index.quarter
    df['month'] = df.index.month
    df['year'] = df.index.year
    df['dayofyear'] = df.index.dayofyear
    df['dayofmonth'] = df.index.day
    df['weekofyear'] = df.index.isocalendar().week

    return df

df = create_features(df)
```

FIGURE 5: CODE PERMETTANT L'AJOUT DE VARIABLES TEMPORELLES

Lorsqu'on affiche la distribution de la consommation en fonction de ces variables on constate qu'il y a bel et bien un effet de saisonnalité.

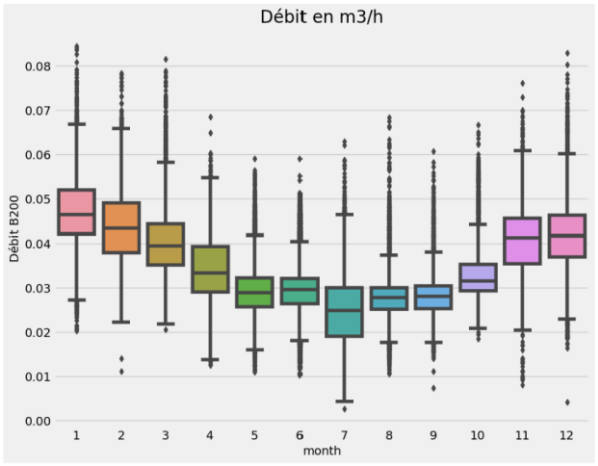

FIGURE 6 : BOITE A MOUSTACHE REPRESENTANT LA REPARTITION DES VALEURS DE DEBITS PAR MOIS

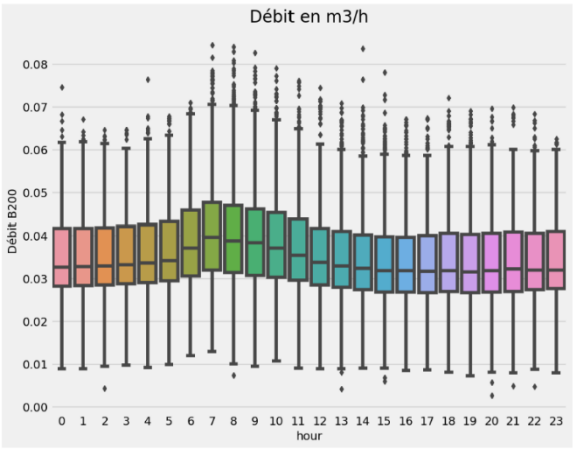

FIGURE 7 : BOITE A MOUSTACHE REPRESENTANT LA REPARTITION DES VALEURS DE DEBITS PAR HEURE

De plus, j'ai rajouté un « décalage » ; une variable qui prend la valeur de la consommation de vapeur, il y a un an / deux ans / trois ans, jour pour jour.

Entrainement du modèle :

J'ai utilisé le modèle XGBRegressor pour effectuer mes prédictions. Ce modèle est un algorithme de gradient boosting de type arbre de décision. Il s'agit d'un modèle de type "ensembliste", qui consiste en l'assemblage de plusieurs modèles plus simples pour créer un modèle plus complexe et plus précis.

Dans mon modèle, j'ai utilisé plusieurs hyperparamètres pour améliorer les performances du modèle. J'ai fixé le nombre d'estimateurs à 1700 et j'ai utilisé une early stopping round à 1700 pour éviter le surapprentissage. J'ai également défini la profondeur maximale de l'arbre à 3 et ai choisi un taux d'apprentissage de 0,01.

Pour évaluer le modèle, j'ai choisi d'utiliser la métrique RMSE ; la moyenne de l'erreur quadratique entre la valeur prédite et la valeur réelle. Plus le RMSE est faible, plus le modèle est précis.

```
[0]     validation_0-rmse:0.46003       validation_1-rmse:0.45992
[100]   validation_0-rmse:0.16853       validation_1-rmse:0.16945
[200]   validation_0-rmse:0.06198       validation_1-rmse:0.06337
[300]   validation_0-rmse:0.02337       validation_1-rmse:0.02474
[400]   validation_0-rmse:0.01010       validation_1-rmse:0.01090
[500]   validation_0-rmse:0.00640       validation_1-rmse:0.00625
[600]   validation_0-rmse:0.00562       validation_1-rmse:0.00477
[700]   validation_0-rmse:0.00542       validation_1-rmse:0.00431
[800]   validation_0-rmse:0.00531       validation_1-rmse:0.00416
[900]   validation_0-rmse:0.00524       validation_1-rmse:0.00407
[1000]  validation_0-rmse:0.00518       validation_1-rmse:0.00405
[1100]  validation_0-rmse:0.00513       validation_1-rmse:0.00401
[1200]  validation_0-rmse:0.00508       validation_1-rmse:0.00400
[1300]  validation_0-rmse:0.00504       validation_1-rmse:0.00399
[1400]  validation_0-rmse:0.00500       validation_1-rmse:0.00397
[1500]  validation_0-rmse:0.00496       validation_1-rmse:0.00397
[1600]  validation_0-rmse:0.00493       validation_1-rmse:0.00397
[1699]  validation_0-rmse:0.00490       validation_1-rmse:0.00396
```

FIGURE 8 : EVOLUTION DU SCORE SUR LE TRAIN SET ET LE TEST SET

On obtient l'erreur la plus faible sur le set de test après 1700 itérations, au-delà le score se dégrade. Validation_0 étant les résultats sur les données d'entrainement et validation_1 sur les données de test. On obtient un RMSE de 0.00396.

Voici la prédiction sur le test set comparée aux données réelles :

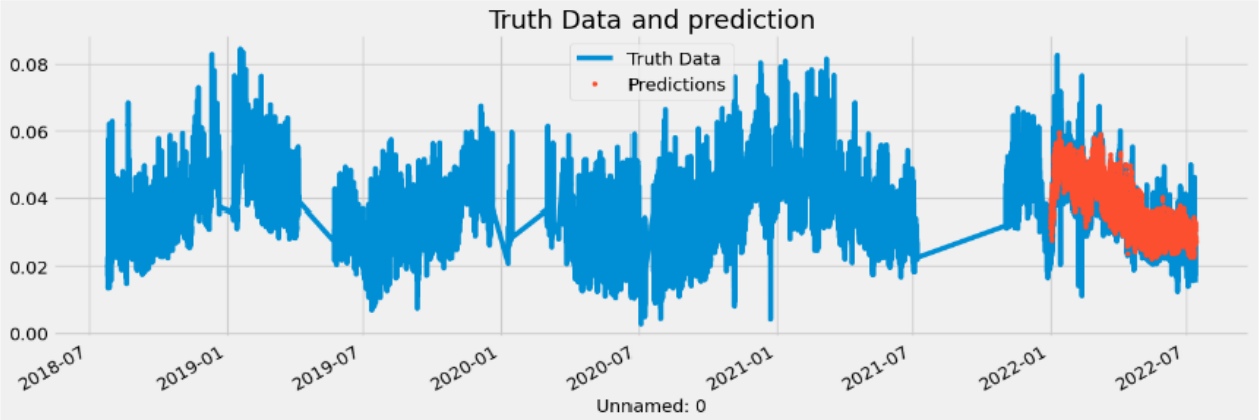

FIGURE 9 : VISUALISATION DE LA PREDICTION PAR RAPPORT AUX DONNEES REELLES

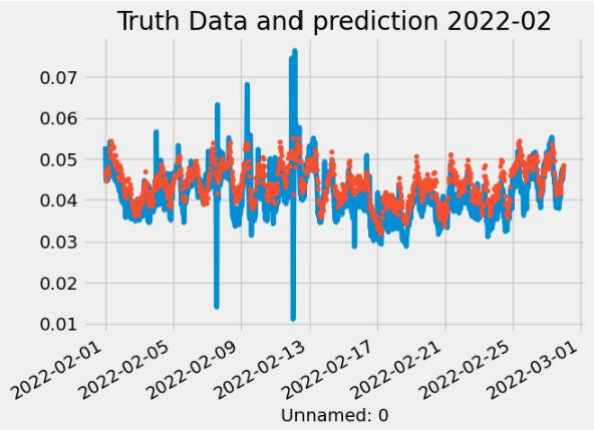

FIGURE 10 : VISUALISATION DE LA PREDICTION PAR RAPPORT AUX DONNEES REELLES

Ainsi, pour notre modèle on arrive à un score R2 de 0,817. Un score R2 élevé indique que le modèle est capable d'expliquer une grande partie de la variance de la variable cible.

```
r2_score(y_test, y_pred)
```

```
0.8168440943736531
```

FIGURE 6 : SCORE R2

L'objectif initial du projet était d'effectuer une prédiction pour la comparer à la valeur relevée.

Avec ce modèle, j'ai pu extraire « l'importance des variables » qui permet de déterminer l'importance de chaque variable dans la prédiction du modèle. Cela permet de déterminer quelles sont les variables avec le plus d'importance dans la consommation d'énergie.

Cela permet de savoir sur quoi il faut agir en priorité lorsque l'on veut effectuer des économies d'énergies.

Conclusion :

Pour réaliser ce projet, je me suis formé en autodidacte. J'ai pu également demander conseil à mon professeur de mathématiques et d'informatique, M. EL-HADI Rebaa. A ce stade, j'ai présenté le projet à mon équipe. Les personnes de mon service n'étant pas familières avec l'intelligence artificielle, j'ai pu faire découvrir l'étendue des projets réalisables avec cet outil. Après réflexion, nous avons estimé qu'il était plus profitable d'effectuer un programme capable de prédire les coefficients de performance des différentes pompes à chaleur pour faire fonctionner en permanence celle qui possède le meilleur rendement que de poursuivre ce projet. Ce projet est en cours et d'après les études que j'ai consultées, des projets similaires ont été réalisés et les meilleurs résultats ont été obtenus avec des réseaux de neurones. Je suis donc en train de me former au Deep Learning pour pouvoir réaliser ce projet.