

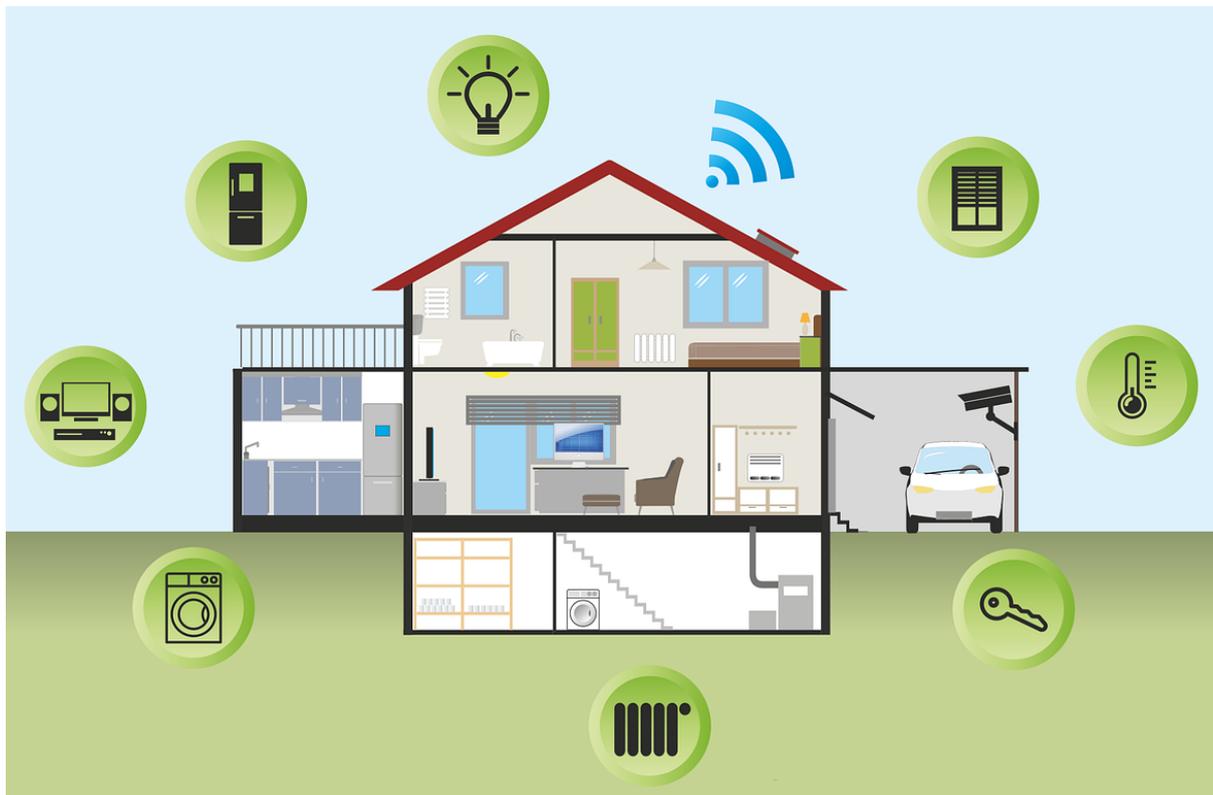


## Projet gestion d'énergie

---

# Étude de l'influence de la consommation de gaz d'une maison liée à la météo.

---



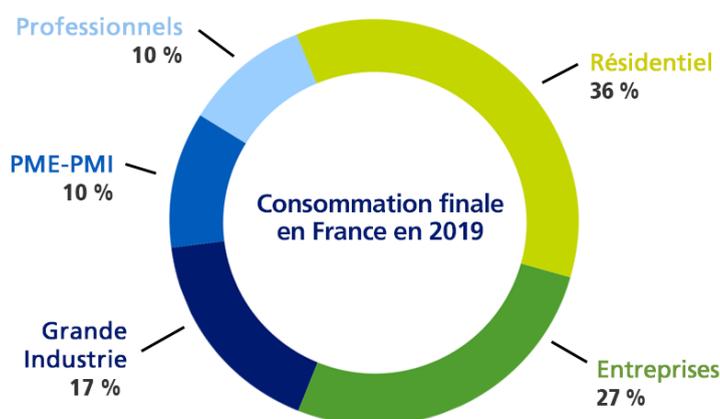
<b>I. Introduction</b>	<b>3</b>
<b>II. Analyse des données</b>	<b>4</b>
a) <b>Récupération des données</b>	<b>4</b>
b) Traitement des données	4
c) Création d'un decision tree	6
<b>III. Conclusion</b>	<b>8</b>
<b>IV. Annexe</b>	<b>9</b>
a) Récupération des données et création d'un fichier excel	9
b) Traitement des données	10
c) Création du decision Tree	11

## I. Introduction

Le secteur du bâtiment résidentiel-tertiaire en France a un impact environnemental non négligeable. Ce secteur, à l'origine de 20% des émissions de CO<sub>2</sub> en France, est le second poste d'émission derrière les transports. Il est également à l'origine de près de 1/3 de la consommation d'énergie finale. Il s'agit donc d'un enjeu majeur des politiques de lutte contre le réchauffement climatique.

Néanmoins, il ne faut pas oublier que le réchauffement climatique va influencer sur les consommations énergétiques des ménages. Des prédictions faites par Météo France montrent que les températures moyennes pourraient atteindre une augmentation de +4°C à l'horizon 2100 si aucune politique de réduction des émissions de gaz à effet de serre n'est mise en place. Cette augmentation de température sera aussi accompagnée de grande variation de température qu'elle soit très chaude en été ou très froide en hiver.

Cette hausse de température risque d'entraîner une hausse de la consommation en climatisation en été et une baisse de la consommation de chauffage en hiver.



La consommation d'électricité par typologie de consommateurs en France en 2019

Source RTE - bilan électrique 2019

© EDF

On souhaite s'intéresser à l'étude de la consommation de gaz dans une maison connectée en fonction de la météo. Cette maison possède un capteur permettant de déterminer la consommation de gaz en m<sup>3</sup> avec un pas de temps de 15min. Cette consommation de gaz est utilisée pour le chauffage de la maison et pour l'eau chaude. Cependant, le capteur ne permet pas de différencier la quantité de gaz utilisée pour le chauffage de celui de l'eau

chaude. Au niveau de la météo, on va s'intéresser, notamment à la température extérieure mais aussi regarder l'influence de la pluviométrie sur la consommation de gaz.

On cherchera à montrer que la météo, et principalement la température, influe beaucoup sur la consommation de gaz dans une maison, puis on essaiera de déterminer la quantité de gaz que va devoir fournir une maison en connaissant la météo.

Il peut être intéressant de pouvoir déterminer la quantité d'eau chaude et de chauffage utilisée par une maison en fonction des conditions météo extérieures, notamment pour le développement des réseaux de chaleur urbains qui fournissent de l'eau chaude sanitaire et du chauffage pour des centaines de foyers.

## II. Analyse des données

### a) Récupération des données

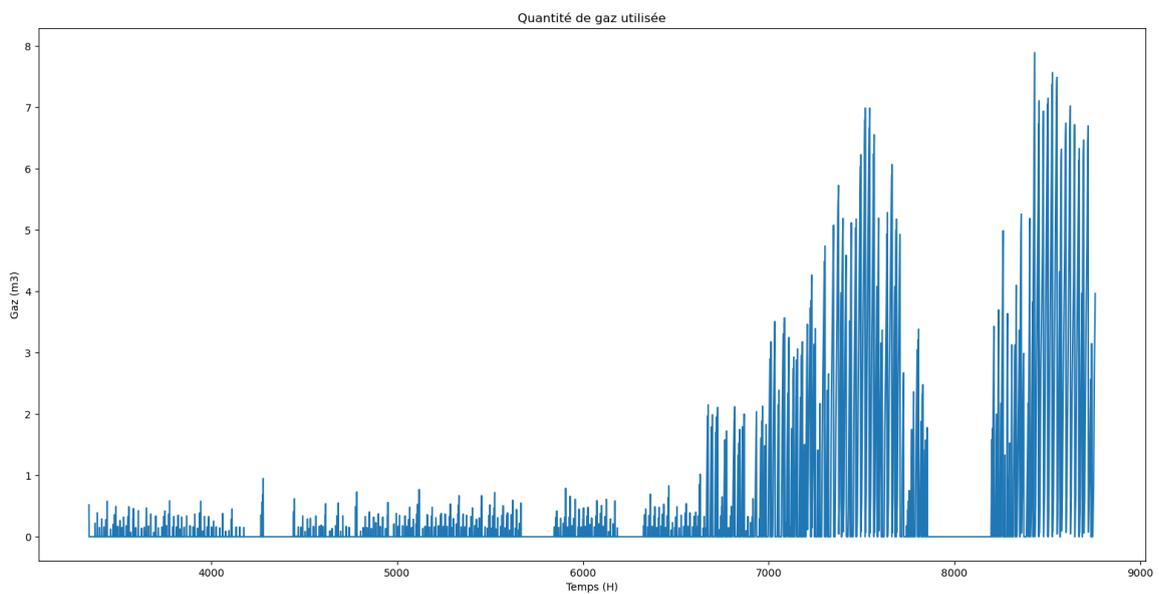
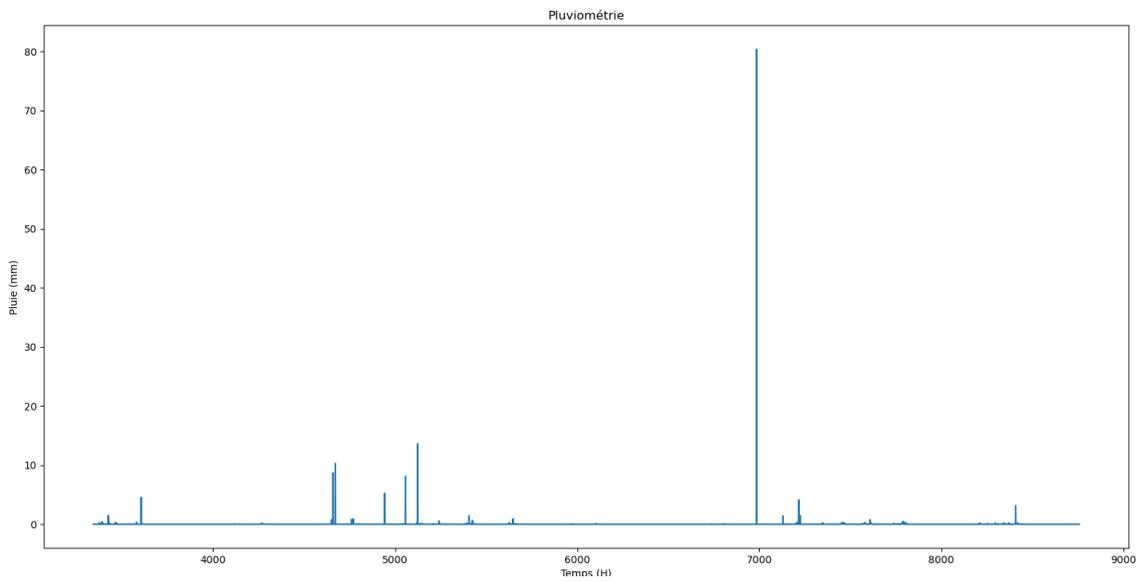
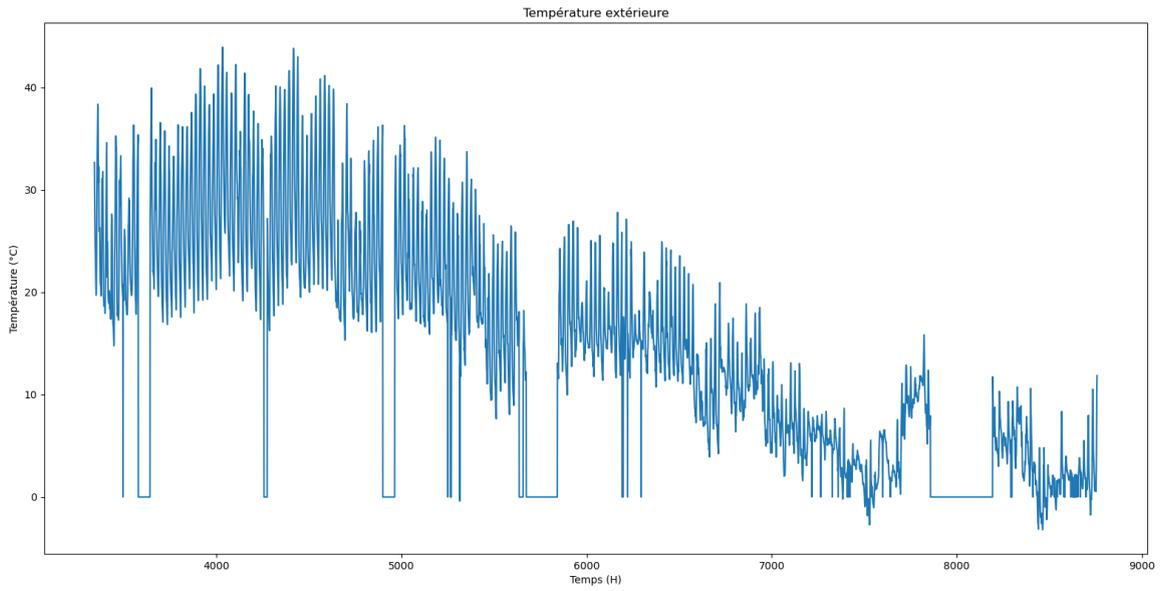
Le projet Smart House a été initié en 2018 afin de fournir les données d'un foyer de 120 m<sup>2</sup> où vit une famille de 5 personnes. Ce projet donne accès à des centaines de points de mesure, accessibles en temps réel via un portail Grafana alimenté avec une base de données Influxdb. Cette maison intelligente est développée sur la base de matériel et de logiciels Open Source.

A l'aide d'un script Python nous récupérons les données qui nous intéressent, c'est-à-dire la consommation de gaz, la température extérieure et la pluviométrie. Puis nous convertissons les données brutes en fichier .csv que nous allons pouvoir exploiter.

### b) Traitement des données

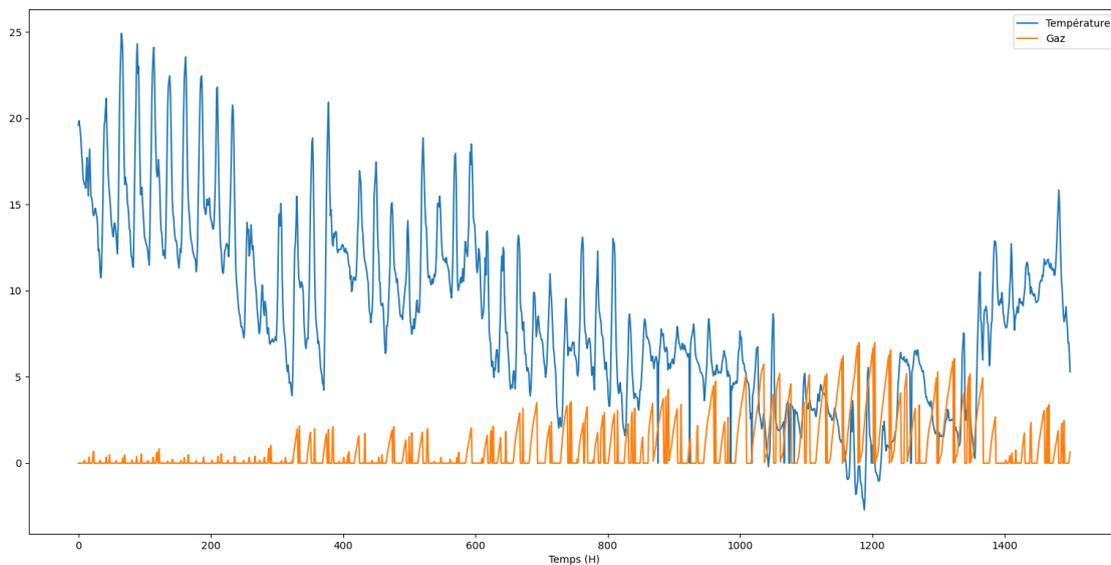
Maintenant que nos données sont en format .csv, nous voulons analyser ces données. Cependant, nous travaillons avec des données réelles qui présentent des défauts, des doublons ou des trous dans les données. Pour pouvoir exploiter ces données il faut donc les traiter pour rendre la comparaison cohérente. Pour cela, nous avons commencé par supprimer les données en double, puis remplacer les données dites NaN (Not a number) par des 0. Ensuite, avec la fonction resample nous avons mis toutes les données sur une même échelle de temps. Nous avons pris des données toutes les heures.

Le capteur permettant de mesurer la quantité de gaz utilisée a été changé en juin dernier, nous avons donc récolté les données de juin à aujourd'hui. Nous obtenons les 3 courbes suivantes :



On remarque qu'à certaine période, les capteurs de température et de gaz ne fonctionnent pas, ils indiquent une valeur nul pendant plusieurs heures d'affilée. On remarque aussi que de juin à début octobre, la quantité de gaz utilisée est plutôt constante et assez faible. Cela est dû au fait qu'il fait chaud et que le chauffage n'est pas allumé. Le gaz est utilisé seulement pour l'eau chaude sanitaire.

Dans la réalité, il est plus intéressant de pouvoir déterminer la quantité de gaz nécessaire lorsqu'on en utilise beaucoup. On va donc s'intéresser uniquement à la période hivernale lorsque le capteur fonctionne. On réduit la période d'étude du 15 octobre au 15 décembre. On obtient les courbes suivantes :



### c) Création d'un decision tree

Un arbre de décision est un modèle de classification en apprentissage supervisé utilisé pour prédire une variable cible en se basant sur des variables explicatives. Il représente les choix qui peuvent être faits à chaque nœud dans l'arbre en utilisant des critères de sélection, lesquels permettent de diviser les données en sous-groupes plus petits et plus homogènes. Chaque nœud de l'arbre représente une décision basée sur une des variables explicatives, tandis que les feuilles de l'arbre représentent les prédictions finales.

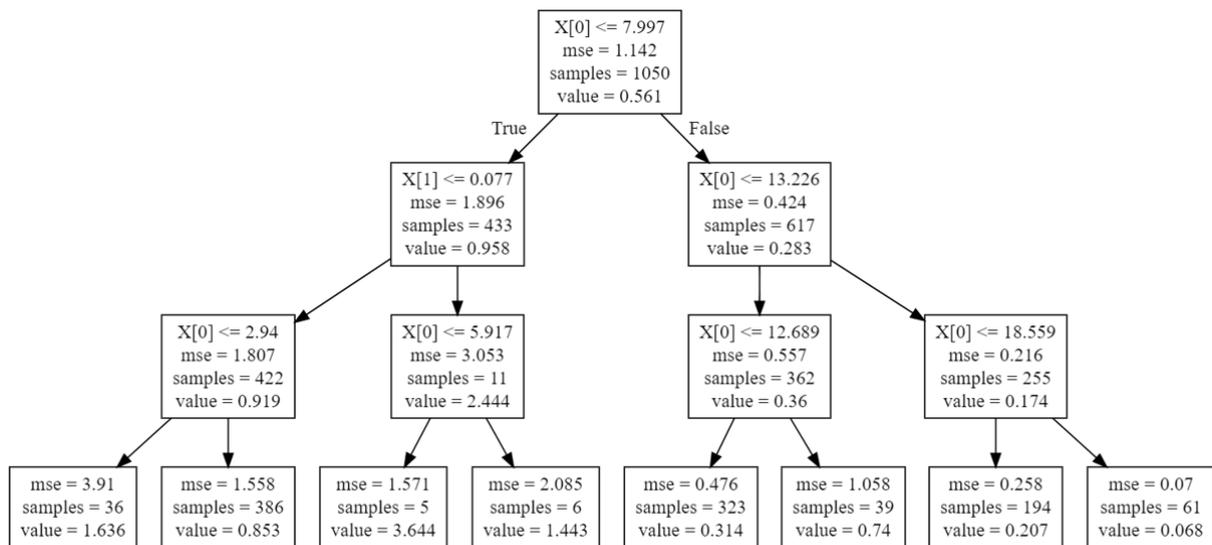
Les arbres de décision sont utilisés en raison de leur simplicité et leur capacité à gérer les interactions complexes entre les variables.

La température et la quantité de gaz utilisées sont des variables continues, c'est pourquoi la méthode de régression paraît plus pertinente que la méthode du Tree classifieur car elle permet d'utiliser ces données.

On choisit une profondeur d'arbre de 3 afin d'avoir de la visibilité et réaliser notre analyse. Les valeurs du gaz sont d'abord séparées en 2 grâce à la valeur de la température extérieure, puis en fonction de la quantité de pluie qui s'écoule au même instant et enfin de nouveau séparées par la température extérieure.

Tableau de correspondance:

X[0]	Température extérieure
X[1]	Pluviométrie
Value	Quantité de gaz utilisée que l'on cherche à prédire
Samples	nombre d'échantillons appartenant au cas étudié sur cette période de temps



Observation:

D'après notre étude, la quantité de gaz utilisée varie entre 0,207 et 3,644 m3 mais la majorité des échantillons sont réparties entre 0,2 et 0,9 m3. Cela signifie que l'utilisation du gaz est assez stable au fur à mesure du mois et qu'il est assez peu utilisé en général en sachant que l'utilisation moyenne d'un foyer est de 1,3 m3.

Pour bien faire il faudrait comparer ces valeurs obtenues à la réalité, cependant nos valeurs étant continues, nos fonctions usuelles ne fonctionnent pas et ne nous permettent donc pas de faire la comparaison.

### III. Conclusion

L'optimisation de la gestion de l'énergie a pour but de réduire la consommation d'énergie et d'atténuer son impact sur l'environnement, tout en améliorant le confort de l'habitation. Nous étions en train d'examiner l'isolation d'une maison connectée à travers l'efficacité du système de chauffage.

Au cours de notre étude, nous avons acquis des compétences en matière de traitement de données pour tirer des conclusions sur la gestion du chauffage de l'habitation. Nous avons également développé des capacités à faire face aux nombreux défis techniques rencontrés, allant de la récupération des données à leur analyse en passant par l'implémentation d'un code en Python.

## IV. Annexe

### a) Récupération des données et création d'un fichier excel

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import urllib
import time
from influxdb import InfluxDBClient
import sys
import csv
import pandas as pd
import matplotlib.pyplot as plt

#####
# SCRIPT SETTINGS
#####
# Set the port where you want the bridge service to run
PORT_NUMBER = 1234
# InfluxDB Server parameters
INFLUXDB_SERVER_IP = '82.65.155.71'
INFLUXDB_SERVER_PORT = 8086
INFLUXDB_USERNAME = 'aleves'
INFLUXDB_PASSWORD = 'Smarthouse62Elab'
INFLUXDB_DB_NAME = 'jeedom'
#####

client = InfluxDBClient(INFLUXDB_SERVER_IP, INFLUXDB_SERVER_PORT, INFLUXDB_USERNAME, INFLUXDB_PASSWORD, INFLUXDB_DB_NAME,ssl=True, verify_ssl=False)

print(client.get_list_database())

client.switch_database('jeedom')

datasheet = client.query('SELECT "value" FROM "jeedom"."autogen"."1768" WHERE time > now() - 365d') # Récupération des données de la maison
```

```
datasheet = client.query('SELECT "value" FROM "jeedom"."autogen"."1768" WHERE time > now() - 365d') # Récupération des données de la maison

exported_data = list(datasheet.get_points())
header_list = list(exported_data[0].keys())

## Création du fichier

with open("C:/Users/erwan/Documents/TP_Smart_System/Temp1.csv", "w", newline='') as fp:
    writer = csv.writer(fp, dialect='excel')
    value_header = header_list[1]
    offset = sum(c.isalpha() for c in value_header)
    print(offset)
    header_list[1:] = ['value']
    print(header_list)
    writer.writerow(header_list)
    for line in exported_data:
        writer.writerow([line[kn] for kn in header_list])

df = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Temp1.csv')

df['time'] = pd.to_datetime(df['time']) # Transforme en temps
df['time'] = df['time'].dt.floor('s') # Enlève les micro seconde
df.drop_duplicates(inplace=True) # Enlève les données en trop (2 fois même donnée)

x = df['time']
y = df['value']
print(df)
```

## b) Traitement des données

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor
import sklearn.metrics as metrics

## Resample des données

# df_gaz = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Gaz_removed.csv', parse_dates=["time"], index_col="time")
# gaz_res = df_gaz.resample("1H").mean()
# gaz_res.to_csv('C:/Users/erwan/Documents/TP_Smart_System/Gaz_res.csv', index=False)
# print(gaz_res)
#
# df_boiler = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Boiler_removed.csv', parse_dates=["time"], index_col="time")
# boiler_res = df_boiler.resample("1H").mean()
# boiler_res.to_csv('C:/Users/erwan/Documents/TP_Smart_System/Boiler_res.csv', index=False)
# print(boiler_res)
#
# df_rain = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Rain_removed.csv', parse_dates=["time"], index_col="time")
# rain_res = df_rain.resample("1H").mean()
# rain_res.to_csv('C:/Users/erwan/Documents/TP_Smart_System/Rain_res.csv', index=False)
# print(rain_res)
#
# df_temp = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Température_removed.csv', parse_dates=["time"], index_col="time")
# temp_res = df_temp.resample("1H").mean()
# temp_res.to_csv('C:/Users/erwan/Documents/TP_Smart_System/Temp_res.csv', index=False)

# df_temp = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Température_removed.csv', parse_dates=["time"], index_col="time")
# temp_res = df_temp.resample("1H").mean()
# temp_res.to_csv('C:/Users/erwan/Documents/TP_Smart_System/Temp_res.csv', index=False)

df_Temp = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Temp_res.csv')
df_Gaz = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Gaz_res.csv')
df_Boiler = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Boiler_res.csv')
df_Rain = pd.read_csv('C:/Users/erwan/Documents/TP_Smart_System/Rain_res.csv')

## On remplace les NaN par des 0

df_Temp = df_Temp.fillna(0)
df_Gaz = df_Gaz.fillna(0)
df_Boiler = df_Boiler.fillna(0)
df_Rain = df_Rain.fillna(0)

## Création de liste

Temp = df_Temp['value'].values.tolist()
Gaz = df_Gaz['value'].values.tolist()
Boiler = df_Boiler['value'].values.tolist()
Rain = df_Rain['value'].values.tolist()

print(len(Temp))
print(len(Gaz))
print(len(Rain))
```

```
# plt.plot(Temp[6342:7842], label='Température extérieure')
# plt.title("Température extérieure")
# plt.xlabel("Temps (H)")
# plt.ylabel("Température (°C)")
# plt.figure()
#
# plt.plot(Gaz[3000:4500])
# plt.title("Quantité de gaz utilisée")
# plt.xlabel("Temps (H)")
# plt.ylabel("Gaz (m3)")
# plt.figure()
#
# plt.plot(Rain[6342:7842])
# plt.title("Pluviométrie")
# plt.xlabel("Temps (H)")
# plt.ylabel("Pluie (mm)")
# plt.figure()
#
# plt.plot(Temp[6342:7842], label='Température')
# plt.plot(Gaz[3000:4500], label='Gaz')
# plt.plot(Rain[6342:7842], label='Pluie')
# plt.xlabel("Temps (H)")
# plt.legend()

Temp_new = np.transpose(Temp[6342:7842])
Gaz_new = np.transpose(Gaz[3000:4500])
Rain_new = np.transpose(Rain[6342:7842])
```

### c) Création du decision Tree

```
data = pd.DataFrame(columns=['Temperature', "Gaz", "Rain"])
data['Temperature'] = Temp_new
data['Gaz'] = Gaz_new
data['Rain'] = Rain_new

print(data)

# Decision Tree

## Ne fonctionne pas parce que variable continue et non discret

# x_train, x_test, y_train, y_test = train_test_split(data[['Temperature', 'Rain']], data[['Gaz']], test_size=0.3, random_state=0)
#
# clf = tree.DecisionTreeClassifier()
# clf = clf.fit(x_train, y_train)
# y_pred = clf.predict(x_test)
# tree.plot_tree(clf)
# with open("tree.dot", 'w') as f:
#     f = tree.export_graphviz(clf, out_file=f)

feature = []

Taille_training = int(0.7*len(Temp_new))
```

```
feature = []

Taille_training = int(0.7*len(Temp_new))

X = []
Y = []

for i in range_(Taille_training):
    feature = [Temp_new[i], Rain_new[i]]
    X.append(feature)
    Y.append(Gaz_new[i])

regr = DecisionTreeRegressor(max_depth=3)
regr.fit(X, Y)
y = regr.predict(X)

with open("tree23.dot", "w") as file:
    f = tree.export_graphviz(regr, out_file=file)
```