GAUTHIER Esteban - TREREMI Romain

Smart System
Miniproject Smart House

Etude pour un dimensionnement optimal de la puissance du compteur électrique

Jérôme Ferrari

2021 / 2022

Objectifs

- Étudier le profil de consommation de la maison pour optimiser le contrat de souscription de puissance électrique.

- Agréger les données de consommation des appareils électriques de la maison pour évaluer la possibilité de passer à un contrat 6 kVA au lieu de 9 kVA.

Mots clés

Smart Home
Puissance électrique
Appareils électroménagers
Analyse de données

Table des matières

Résumé

De nos jours, la technologie “Internet of Things” (IoT) connaît un essor considérable. L’IoT désigne la connexion des objets et des systèmes techniques à Internet, notamment à travers l’agrégation de données et la multiplication des capteurs. Les notions de cloud et d’objets connectés ouvrent le champ des possibles concernant le pilotage intelligent des systèmes et l’optimisation des flux de consommation d’énergie.

Le projet repose sur l’étude d’une *smart house* de 120m^2 hébergeant une famille de 5 personnes. Un ensemble de capteurs a été implémenté donnant accès à près de 340 points de mesure. Les séries de données sont stockées sur InfluxDB et accessibles via l’interface Grafana, ainsi que par Python. L’objectif du projet sera de proposer un script python constituant un système domotique intelligent. En particulier, l’accès à des informations telles que la consommation d’électricité, de gaz, d’eau en fonction de l’occupation de la maison peut conduire à une optimisation des besoins en énergie du foyer.

Notre analyse portera plus particulièrement sur l’étude du profil de consommation de la maison afin d’optimiser le contrat de souscription de puissance électrique. La puissance du compteur électrique, exprimée en kVA, correspond à la puissance maximale que peut délivrer le compteur électrique. Ce seuil doit théoriquement supporter la puissance atteinte lorsque tous les appareils électriques de la maison fonctionnent en même temps. Si la puissance soutirée dépasse le seuil du contrat électrique, le compteur disjoncte d’où la nécessité de calibrer son abonnement de sorte à ne pas être sujet à des coupures régulières. Le tableau ci-dessous précise la puissance souscrite des ménages selon les caractéristiques du logement et les usages observés.

Caractéristiques du logement	Puissance conseillée
Studio ou appartement < 50m² Sans chauffage électrique 1 gros appareil électroménager	3 kVA
Appartement ou maison < 80m² Chauffage électrique 2 gros appareils électroménagers	6 kVA
Appartement ou maison > 80 m² Chauffage électrique et/ou climatisation 2 à 3 gros appareils électroménagers	9 kVA
Maison > 100 m² Chauffage électrique et/ou climatisation 3 gros appareils électroménagers	12 kVA
Maison > 100 m² Chauffage électrique et/ou climatisation Au moins un appareil énergivore (par ex. : pompe à chaleur) Plus de 3 gros appareils électroménagers	15 à 36 kVA

Puissance souscrite recommandée selon les caractéristiques du logement (Source : Total Energies)

En réalité, il est possible d'identifier le contrat de souscription optimal dès lors que nous avons accès aux relevés détaillés de consommation en électricité du logement. L'idée sera d'identifier les pics réellement observés de puissance soutirée sur une période suffisamment longue. Pour ces pics, nous identifierons les équipements responsables de la forte consommation en énergie dans le but de proposer une meilleure utilisation de ces derniers. Par exemple, espacer le fonctionnement des équipements électriques dans la journée permet d'éviter les pics et donc avoir recours à un contrat d'abonnement plus bas. Une gestion optimale de sa consommation d'électricité permet à l'utilisateur de réaliser des économies: le tableau ci-dessous présente le prix croissant de l'abonnement annuel selon la puissance souscrite.

Grille tarifaire - tarifs réglementés EDF - option base

Puissance souscrite	Abonnement annuel TTC	Prix du kWh TTC
3 kVA	104.89 €	0.1558 €
6 kVA	137.64 €	0.1558 €
9 kVA	171.28 €	0.1605 €
12 kVA	205.42 €	0.1605 €
15 kVA	237.67 €	0.1605 €
18 kVA	271.06 €	0.1605 €
24 kVA	340.74 €	0.1605 €
30 kVA	408.40 €	0.1605 €
36 kVA	477.70 €	0.1605 €

Prix à jour au 1er janvier 2022

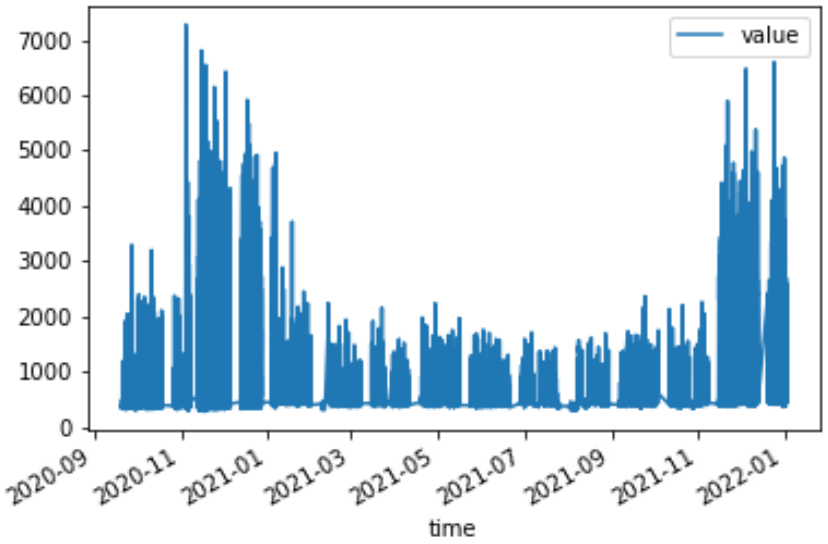
Tarif bleu EDF et son abonnement en option base

I - Acquisition des données

L'objectif du projet est d'évaluer la possibilité pour le logement étudié de passer à un contrat d'abonnement 6 kVA au lieu de 9 kVA. Il faudra évaluer la faisabilité d'un tel changement en prenant en compte le confort des occupants, c'est-à-dire en maintenant une fréquence des coupures de courant proche de 0.

A) Etude du profil de consommation

La consommation d'énergie électrique du foyer est obtenue grâce aux relevés de mesure fournis par le compteur Linky. Nous avons tracé la puissance totale soutirée en kVA sur une période de 1 ans et 4 mois, du 19/09/2020 au 10/01/2022 et avec un pas de temps de l'ordre de 10 secondes. Le résultat de ce relevé est donné dans la figure ci-après.



Puissance soutirée en kVA (Linky) sur une plage du 19-09-2020 au 10-01-2022

On constate que sur la période étudiée, la puissance soutirée ne dépasse pas 2,5 kVA pendant une grande majorité du temps. En effet, sur la période de février à novembre, c'est-à-dire hors période hivernale, le seuil de puissance maximale est relativement bas. Un abaissement du contrat de souscription de 9 kVA à 6 kVA n'aurait donc pas d'impact sur les habitudes des occupants sur cette partie de l'année.
La période considérée pour l'optimisation de la consommation électrique du foyer sera donc restreinte aux mois de Novembre, Décembre et Janvier durant lesquels ont été mesurés les pics de consommation. Nous avons défini un seuil limite de 5,5 kVA à ne pas dépasser en vue du passage à un contrat en 6 kVA. Cette marge de sécurité a pour but d'élaborer un guide d'utilisation des équipements électriques qui ne soit pas trop proche de la puissance maximale autorisée pour éviter un déclenchement régulier du disjoncteur. Les jours pour lesquels on observe un dépassement sont les suivants, avec une indication de l'heure approximative du pic :

liste des jours = ['2020-11-04, 09',
'2020-11-15, 10',
'2020-11-15, 17',
'2020-11-18, 10',
'2020-11-24, 10',
'2020-11-26, 18',
'2020-12-02, 13',
'2020-12-17, 17',
'2021-11-21, 10',
'2021-12-04, 11',
'2021-12-24, 08']

B) Décryptage des pics de consommation

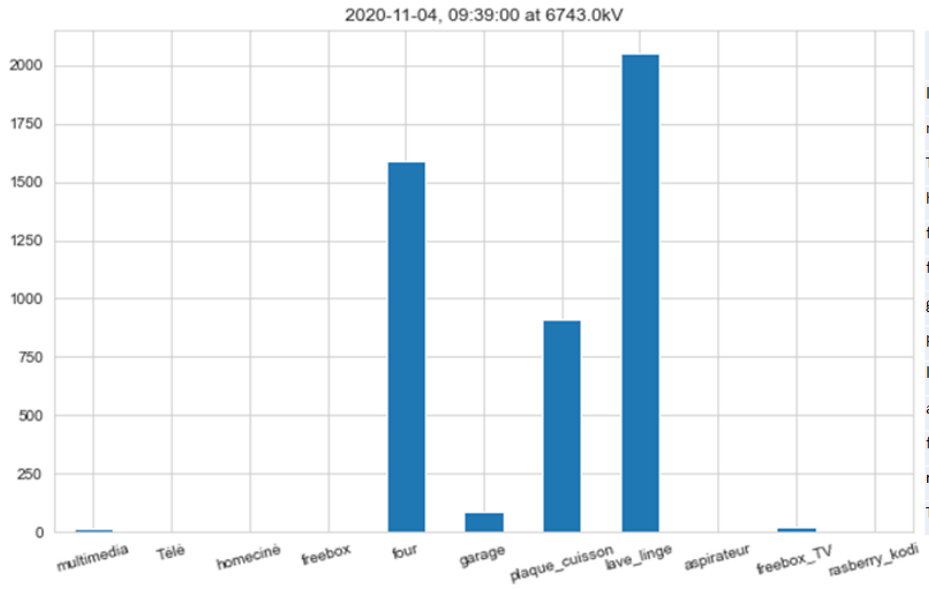
Nous avons décidé de focaliser notre analyse sur 3 des pics détectés précédemment : celui du 04/11/2020 à 09h39 atteignant 6743 kVA, celui du 18/11/2020 à 10h54 atteignant 5987 kVA et celui du 02/12/2020 à 13h47 atteignant 6290 kVA. Ces pics sont les plus significatifs dans le relevé de consommation.

Pour analyser ces 3 pics il a fallu dans un premier temps mettre en forme les données. Tout d'abord il a fallu rééchantillonner chaque mesure sur une même base de temps. Pour ce faire, nous avons utilisé le module Resample de Pandas avec une méthode de moyennage entre 2 valeurs. Cependant, cette méthode implique que certaines valeurs sont inexistantes entre 2 pas de temps. Pour contrer cette perte nous avons interpolé les valeurs en leur attribuant la valeur la plus proche : cette fonction est aussi fournie par Pandas.

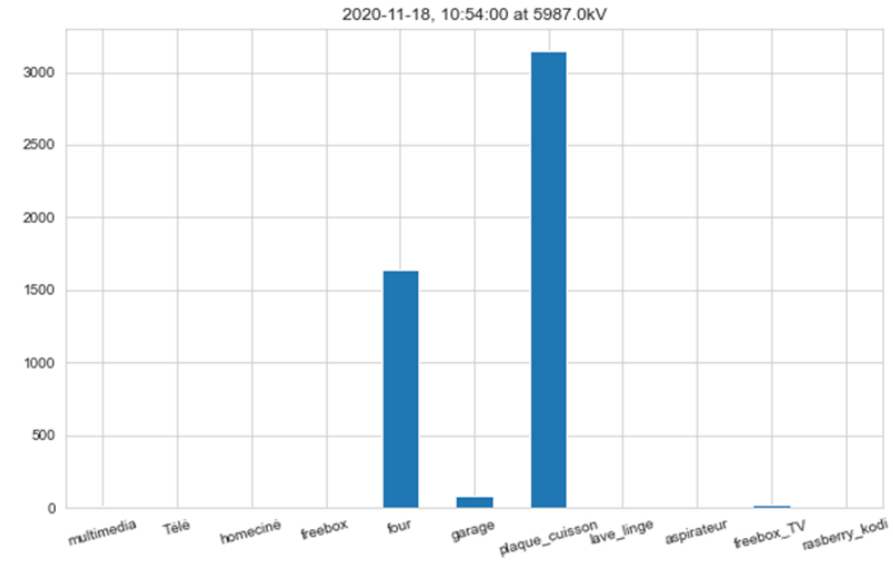
time	linky	multimedia	Télé	homeciné	freebox	four	garage	plaque_cuisson	lave_linge	aspirateur	freebox_TV	rasberry_kodi
2020-09-23 05:35:00	609.640000	14.1	0.000	0.550000	0.0	0.0	72.0	864.0	55.0	2.9	17.6	0.0
2020-09-23 05:36:00	609.640000	14.1	0.000	0.550000	0.0	0.0	70.0	864.0	0.0	2.9	17.6	0.0
2020-09-23 05:37:00	609.640000	14.1	0.000	0.566667	0.0	0.0	71.0	864.0	97.0	2.9	17.6	0.0
2020-09-23 05:38:00	609.640000	14.1	29.780	0.566667	0.0	0.0	72.0	864.0	118.0	2.9	17.6	0.0
2020-09-23 05:39:00	609.640000	14.1	29.780	0.550000	0.0	0.0	71.0	864.0	23.0	2.9	17.6	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
2021-12-31 14:13:00	2138.750000	17.8	4.425	0.560000	3.7	1597.0	92.0	0.0	50.0	2.7	17.7	0.8
2021-12-31 14:14:00	3434.615385	17.8	4.425	0.550000	3.7	59.0	91.0	0.0	50.0	2.7	17.7	0.8
2021-12-31 14:15:00	3987.771659	17.8	4.425	0.550000	3.7	1563.0	91.0	0.0	50.0	2.7	17.7	0.8
2021-12-31 14:16:00	2744.000000	17.8	4.425	0.600000	3.7	1572.0	91.0	0.0	50.0	2.7	17.7	0.8
2021-12-31 14:17:00	2577.333333	17.8	4.425	0.600000	3.7	0.0	90.0	0.0	50.0	2.7	17.7	0.8

Dataset rééchantillonné

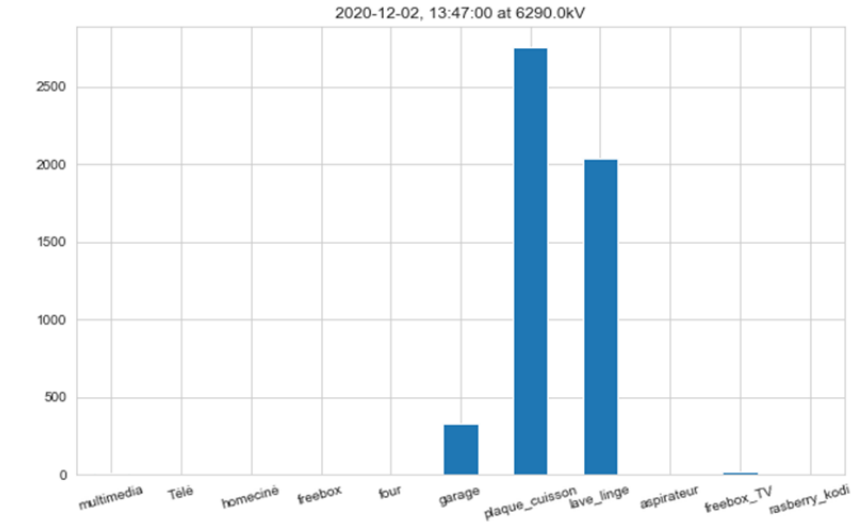
Le détail des consommations par appareil électrique à ces instants est donné sur les graphes ci-dessous :



	2020-11-04 09:39:00
linky	6743,333333
multimedia	10,8
Télé	0
homeciné	0,6
freebox	0
four	1593
garage	83
plaque_cuisson	910
lave_linge	2050
aspirateur	3
freebox_TV	17,7
rasberry_kodi	0,8
Total	4668,9

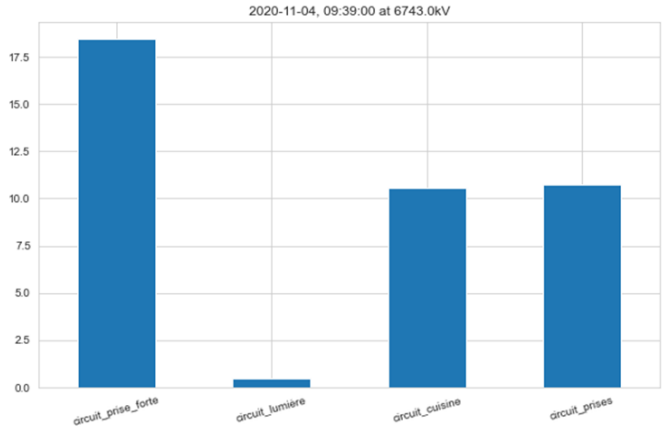


	2020-11-18 10:54:00
linky	5986,8
multimedia	11,8
Télé	0
homeciné	0,57
freebox	0,8
four	1635
garage	76
plaque_cuisson	3144
lave_linge	0
aspirateur	2,9
freebox_TV	17,7
rasberry_kodi	0
Total	4888,8

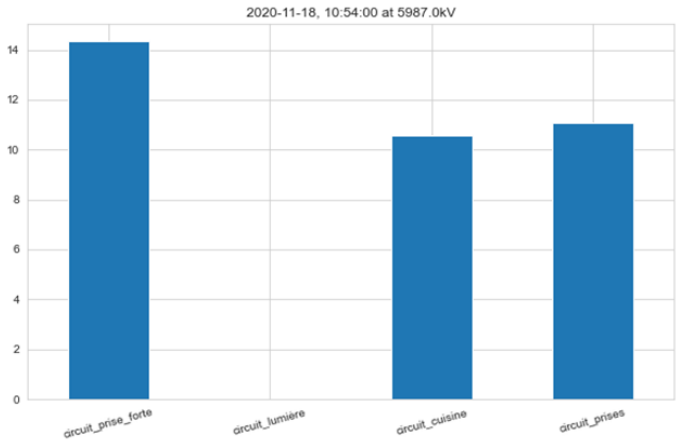


	2020-12-02 13:47:00
linky	6290,43
multimedia	12,6
Télé	0
homeciné	0,55
freebox	3,7
four	0
garage	329
plaque_cuisson	2750
lave_linge	2039
aspirateur	2,9
freebox_TV	17,7
rasberry_kodi	2,5
Total	5157,95

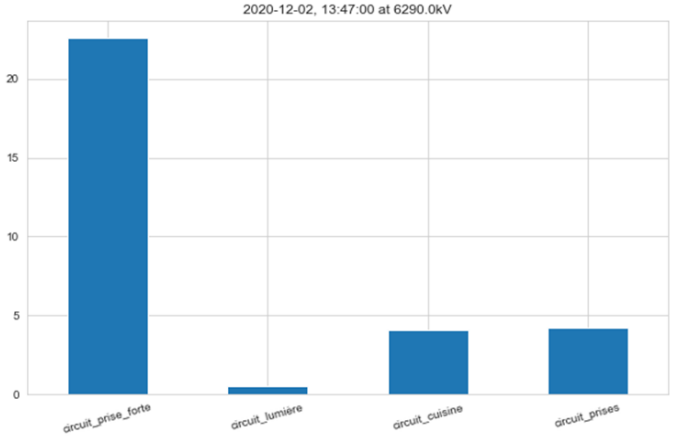
On constate globalement que les maximums de puissance soutirée pour ce foyer interviennent lorsque 2 voire 3 équipements à haute consommation sont mis en marche simultanément. Parmi ces équipements, on identifie le four, les plaques de cuisson et le lave linge. Le garage joue également un rôle non négligeable dans la consommation du foyer, ce qui est dû au fonctionnement d'imprimantes 3D. Les autres équipements (multimédia, Freebox…) ont un poids minoritaire et constituent une consommation résiduelle en fond.
La maison étant aussi équipée avec des capteurs de courant sur les différentes prises, il est possible d'appliquer la méthode explicitée ci-dessus sur ces données.



	2020-11-04 09:39:00
linky	6743,3
circuit_prise_forte	18,43
circuit_lumière	0,5
circuit_cuisine	10,54
circuit_prises	10,75



	2020-11-18 10:54:00
linky	5986,84
circuit_prise_forte	14,33
circuit_lumière	0
circuit_cuisine	10,55
circuit_prises	11,08

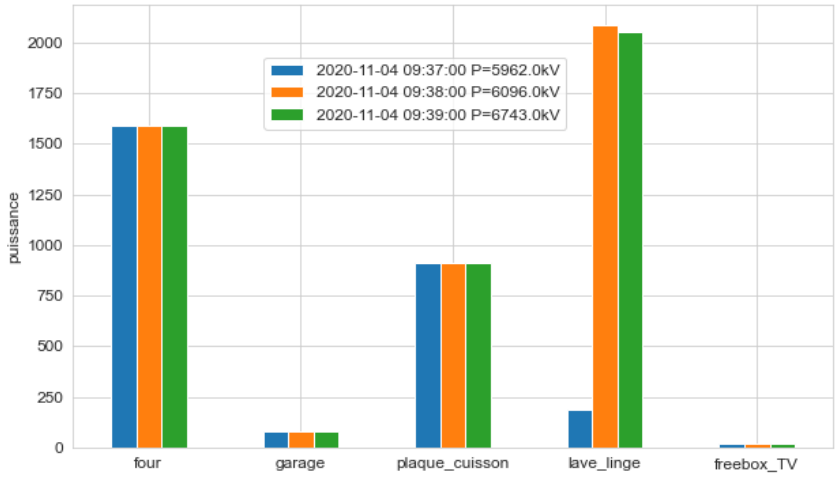


	2020-12-02 13:47:00
linky	6 290,43
circuit_prise_forte	22,58
circuit_lumière	0,48
circuit_cuisine	4,05
circuit_prises	4,21

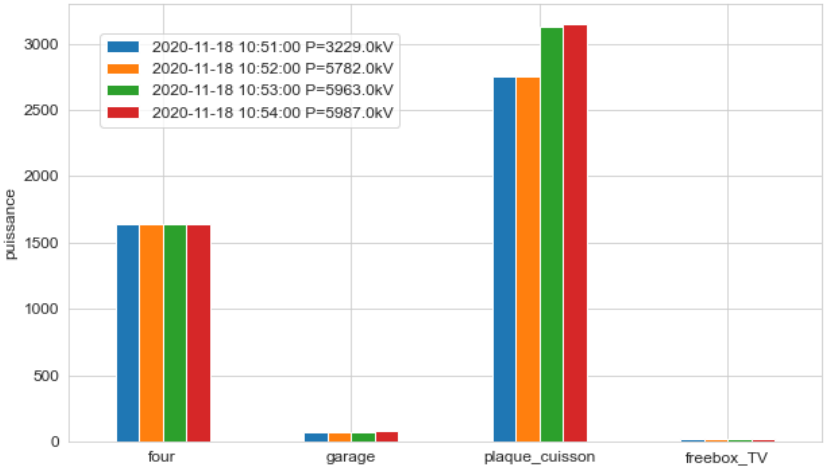
Ces graphiques montrent que les pics de puissance sont dus à un fort courant dans le circuit 'prise forte'.

Pour décrypter plus finement les pics de consommation, nous avons observé la puissance de chaque appareil allumé avec une puissance de plus de 20 W, 4 minutes avant que la puissance du Linky atteigne 6 kVA.

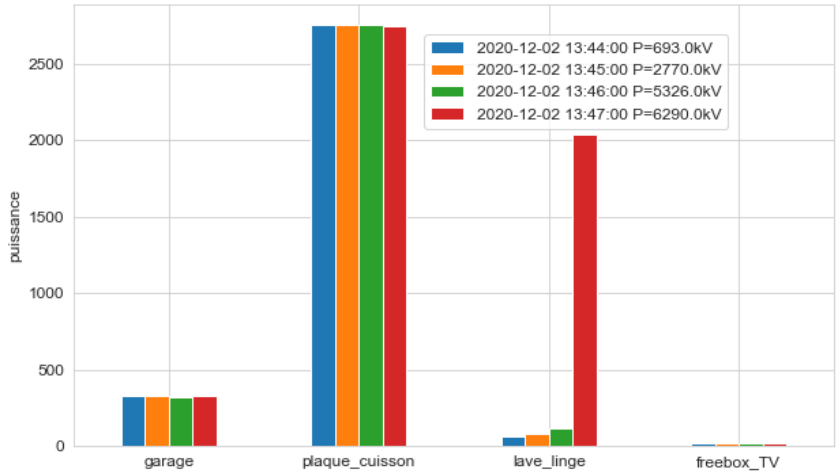
04 nov. 2020 :



18 nov. 2020 :



02 dec. 2020 :



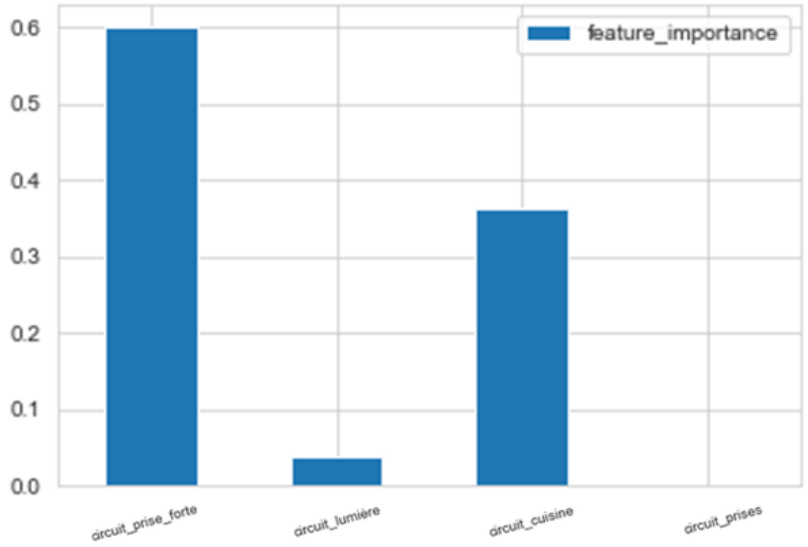
On dénote que les pics sont hautement corrélés à une hausse de la consommation du lave linge qui augmente avec un delta de quasiment 2 kW pour le 04 novembre et le 02 décembre. Pour le 18 novembre, on remarque que c'est l'augmentation due à la plaque de cuisson avec le four allumé qui fait que l'on bascule au-delà du seuil.

II - Traitement des données grâce au machine learning

Maintenant que nous avons identifié manuellement les pics de consommation du foyer et que nous avons une première idée des causes amenant à ces scénarios, nous allons tâcher de vérifier nos hypothèses à l'aide de méthodes de machine learning.
Plus particulièrement, dans cette section nous allons appliquer une méthode de machine learning afin de déterminer quels équipements électroménagers ont le plus d'influence sur les pics de consommation. Nous calculerons pour cela les *feature importances* des dataset en input par rapport au label de notre problème qui est la consommation électrique totale du foyer. Ce label a été préalablement discrétisé de la manière suivante : si la puissance est au dessus de 5,5 kVA la valeur vaut 1 sinon 0.
Notre problème est donné avec le label, nous pouvons donc appliquer une méthode dite supervisée. Nous disposons d'un set de données continues, c'est pourquoi nous appliquerons une méthode de classification appelée DecisionTreeRegressor.

A) Feature importances : prises de courant

Dans une première partie, on considère en entrée de notre algorithme les données de consommation par prise de courant.
On dispose donc de 4 features que nous allons essayer de classer selon leur poids dans la consommation totale : *circuit prises fortes, circuit lumière, circuit cuisine, circuit prises*.
Le graphe donnant le calcul des feature importances est le suivant :



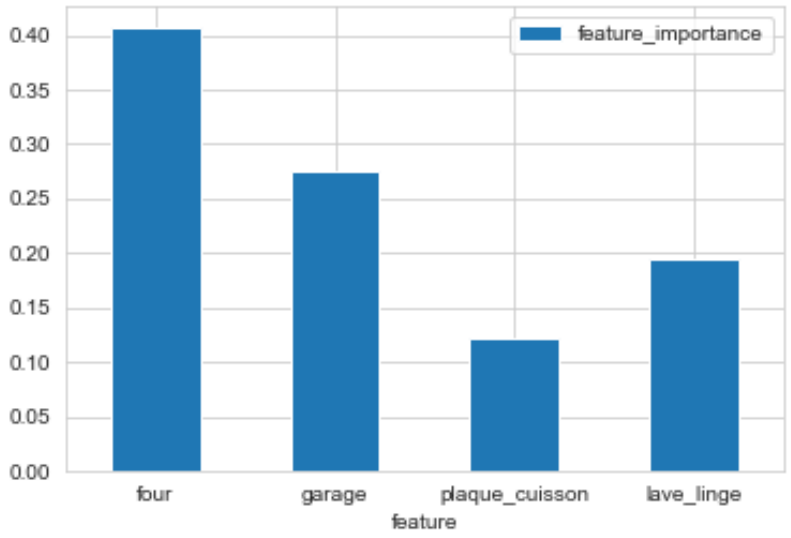
On a donc la hiérarchie suivante :

circuit prises fortes > circuit cuisine > circuit lumière > circuit prises

Cela confirme l'influence majoritaire des équipements à haute puissance branchés sur les prises fortes dans le calcul de la consommation totale.

B) Feature importances : équipements électriques

Pour aller plus loin, nous appliquons la même méthode en considérant comme features les consommations des équipements électriques. Cette méthode consiste à sélectionner les appareils les plus importants c'est-à-dire ayant un maximum de puissance atteinte supérieur à 1 kW. Le dataset contenant les features sélectionnées est ensuite entraîné avec notre modèle de classification.
Sur la dizaine d'équipements électroménagers pour lesquels un relevé de consommation est effectué, seuls 4 ont une importance significative à la suite du calcul de l'algorithme : le four, le garage (imprimantes 3D), les plaques de cuisson et le lave linge. Leurs valeurs de feature importances respectives sont tracées sur le graphe ci-dessous :



Finalement, la méthode de machine learning utilisée confirme la forte influence des équipements à haute puissance sur la consommation totale, et ce sur une période d'environ 1 an et demi. Sur l'ensemble des pics observés, le four, le garage, le lave linge et les plaques de cuisson ont un impact important et sont à la source des pics de puissance.
Pour conclure sur la méthode de machine learning, nous avons vu que sans une analyse physique (partie I.B), il aurait été impossible d'établir une étude par *feature importances*. En effet, il a fallu établir un seuil de puissance pour éliminer les features non pertinentes sinon l'algorithme ne converge pas vers des solutions réalistes. De plus, l'entraînement du modèle voit son efficacité limitée par le grand écart entre le nombre de 1 et de 0 dans notre label : sur les 87841 valeurs, seules 16 prennent la valeur 1.

Conclusion

L'étude menée sur la consommation de cette smart house révèle l'importance respective des équipements électroménagers sur la consommation totale du foyer. Sur près de 70% de la période prise en considération (Septembre 2020 à Janvier 2022), la puissance soutirée par le compteur Linky ne dépasse pas le seuil de 3 kVA. Le reste du temps, un besoin en énergie plus important est requis pour le foyer. Nous avons identifié 10 jours pour lesquels étaient relevées des puissances soutirées sur le réseau supérieures à 5,5 kVA. Pour chacun de ces pics de puissance, nous avons décrypté via 2 méthodes différentes leurs compositions et les facteurs amenant à une telle consommation. Il en ressort qu'une mise en marche simultanée des équipements électroménagers à haute puissance en est la cause principale.
Il convient donc d'espacer l'utilisation des équipements pour éviter d'atteindre des puissances soutirées aussi hautes. Par exemple, différer l'utilisation du lave-linge avec celle du four et des plaques de cuisson contribuerait à lisser la courbe de consommation électrique du ménage. En mettant en place des pratiques de mise en fonctionnement optimisée des équipements, il sera possible de passer à un abonnement 6 kVA et donc de réaliser des économies significatives.

En complément, il est possible de réduire ses factures d'électricité grâce à des pratiques éco-responsables. Par exemple, selon un article de "Maîtriser mon énergie", pour économiser sur la consommation du lave-linge, il existe 3 facteurs:

- La classe énergétique

Elle indique la consommation d'énergie théorique du lave-linge. Elle se décompose en 7 niveaux, de A+++ à D. AAA+ est la classe énergétique la plus économique, D la plus énergivore.

- la température de l'eau

Plus la température de l'eau est élevée, plus la consommation d'électricité du lave-linge est importante.

- La vitesse du cycle

Une vitesse rapide entraîne une consommation d'électricité élevée sur une durée moindre. À l'inverse, une vitesse plus faible engendrera un cycle plus long donc moins énergivore.

Références

https://www.totalenergies.fr/particuliers/electricite/compteur-electrique/comment-choisir-la-puissance-d-un-compteur-electrique?gclid=CjwKCAiAz--OBhBIEiwAG1rIOrFYanfjY9eLMxU22QxTdGtMyVChkVScUlbD-Kfc2a3nblCz0Iq5mBoCe7sQAvD_BwE

https://www.fournisseurs-electricite.com/edf/abonnement

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.resample.html

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

https://www.maitriser-mon-energie.fr/consommation-delectricite-du-lave-linge-tout-savoir/

Annexe

Capteurs nécessaires pour le projet

Linky (puissance consommée en kVA): ID 1503
Capteurs de consommation des appareils électriques de la maison (puissance en kW) :

- TV : 1372
- Freebox : 1377
- Home Ciné : 1382
- Four : 2091
- Machine à laver : 2099
- Robot aspi : 1387
- Freebox TV : 1392
- Raspberry kodi : 1666 (multimédia)
- Multimédia : 1673
- Garage : 2093
- Plaques cuisson : 2095
- Lave vaisselle : 2097
- PZ puissance générale : 5224

Capteurs de courant pour remonter à la totalité de la conso (frigo…)

- prise forte 324
- lumière 326
- cuisine 328
- prises 330
- linky général 1505
- PZEM général courant 5230

Script Python

```
#!/usr/bin/env python
# coding: utf-8

from http.server import BaseHTTPRequestHandler, HTTPServer
import urllib
import time
from influxdb import InfluxDBClient
import sys
import seaborn as sns
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score

Inputs:
path = 'C:\\Users\\eg260905\\Desktop\\ENSE3\\3A\\smart_system\\'

############################
#    SCRIPT SETTINGS
############################
# Set the port where you want the bridge service to run
PORT_NUMBER = 1234
# InfluxDB Server parameters
INLUXDB_SERVER_IP = '192.168.1.210'
INLUXDB_SERVER_PORT = 8086
INFLUXDB_USERNAME = 'eleves'
INFLUXDB_PASSWORD = 'SmarthouseG2Elab'
INFLUXDB_DB_NAME = 'jeedom'
############################

client = InfluxDBClient(INLUXDB_SERVER_IP, INLUXDB_SERVER_PORT,
INFLUXDB_USERNAME, INFLUXDB_PASSWORD, INFLUXDB_DB_NAME)
client.switch_database('jeedom')

#### Timestamps in seconds ###########"
timestamp_end= 1641805072
jour =86400
annee = 365*jour
```

```
timestamp_start=timestamp_end-(annee*2)
dt_object_start = datetime.fromtimestamp(timestamp_start)
print(dt_object_start)
dt_object_end = datetime.fromtimestamp(timestamp_end)
print(dt_object_end)


query = 'SELECT "value" FROM "jeedom"."autogen"."5229" WHERE time >= $start_time
AND time < $end_time'
bind_params = {'end_time': str(dt_object_end), 'start_time': str(dt_object_start)}
result = client.query(query, bind_params=bind_params)


ids = {}
ids['linky'] = '1503'
ids['Télé']= '1372'
ids['homeciné']= '1382'
ids['freebox']= '1377'
ids['four']= '2091'
ids['garage']= '2093' #imprimante_3D
ids['plaque_cuisson']= '2095'
ids['lave_linge']= '2099'
ids['lave_linge']= '2099'
ids['aspirateur']= '1387'
ids['freebox_TV']= '1392'
ids['rasberry_kodi']= '1666'
ids['multimedia']= '1673'
ids['circuit_prise_forte']= '324'
ids['circuit_lumière']= '326'
ids['circuit_cuisine']= '328'
ids['circuit_prises']= '330'
ids['linky_gene']= '1505'
ids['PZEM_gene']= '5230'

result_sets = {}
for name in ids.keys():
   result_sets[name]= {'time':[],'value':[]}


bind_params = {'end_time': str(dt_object_end), 'start_time': str(dt_object_start)}
for id_name in ids.keys():
   query_name = 'SELECT "value" FROM "jeedom"."autogen".'+'"'+ids[id_name]+'"
'+'WHERE time >= $start_time AND time < $end_time'
```

```
    resultset = client.query(query_name, bind_params=bind_params)
    for result in resultset:
        for line in result:
            temps = line['time'].split('T')
            date = temps[0]
            heure = temps[1].replace('Z','')
            heure = heure.split(':')
            date = date.split('-')
            #.strftime("%Y-%m-%d, %H:%M:%S")
            d =
datetime(int(date[0]),int(date[1]),int(date[2]),int(heure[0]),int(heure[1]),int(float(heur
e[2])),0)
            result_sets[id_name]['time'].append(d)
            result_sets[id_name]['value'].append(line['value'])


mergeData=[]
for col_name in result_sets.keys():
    df = pd.DataFrame.from_dict(result_sets[col_name])
    df.columns = ['time '+col_name,col_name]
    mergeData.append(df)


df_pristine=pd.concat(mergeData, axis=1, join='inner')



data_linky = pd.DataFrame.from_dict(result_sets['linky'])
six_kV = {'time':[],'value':[]}
for k in range(len(data_linky)):
    if data_linky['value'][k]>5500:
        six_kV['time'].append(data_linky['time'][k])
        six_kV['value'].append(data_linky['value'][k])

'''
data_linky.plot(x='time',y='value')
plt.savefig(path+'study_range.png')
'''


#### Function ###
def get_sample_mean_df(sensor_data,time_step='1T'):
    '''
    Reample data acquired with a given time step.
```

```
    This function uses the resample module from pandas with a mean method.
    If data has 'nan' value due to the mean method, the algorithm interpolate the value
with the nearest measure
    inputs:
        - sensor_data: dict containing the measurements with its acquisition time
        - time_setp: str 1T = 1 minutes
    outputs:
        - dataframe: the sensor data resampled
    '''
    df = pd.DataFrame.from_dict(sensor_data)
    df.set_index('time', inplace=True)
    df.ffill()
    resampled_df=df.resample(time_step).mean()
    resampled_df.columns = [col_name]
    resampled_df.interpolate(method='nearest', inplace=True)
    return resampled_df

df=pd.concat(mergeData, axis=1, join='inner')

"""df.to_excel(path+'donne.xlsx')"""

list_appareil =
['linky','multimedia','Télé','homeciné','freebox','four','garage','plaque_cuisson','lave_linge','
aspirateur','freebox_TV','rasberry_kodi']
list_circuit = ['linky','circuit_prise_forte','circuit_lumière','circuit_cuisine','circuit_prises']

mergeData=[]
for col_name in list_appareil:
    mergeData.append(get_sample_mean_df(result_sets[col_name],'1T')) # Resample the
data
df_appa=pd.concat(mergeData, axis=1, join='inner') # concatenate the resampled data
into 1 dataframe with the same time index

# Get the values of the linky above 6 kV and creates 1 array with 1 if <6kV or 0
linky_binary = []
six_kV = []
for k in range(df_appa.shape[0]):
    if df_appa['linky'][k]>5500:
        print(round(df_appa['linky'][k]),'kV  at ',df_appa.index[k].strftime("%Y-%m-%d,
%H:%M"))
        six_kV.append(df_appa.index[k])
        linky_binary.append(1)
    else:
```

```
        linky_binary.append(0)

jours = {}
for date_six_kV in six_kV:
    j = date_six_kV.strftime("%Y-%m-%d, %H")

    if (j in jours.keys())==False:
        jours[j] = {'value':df_appa.loc[date_six_kV]['linky'],'index':date_six_kV}
    else:
        if df_appa.loc[date_six_kV]['linky']>jours[j]['value']:
            jours[j] = {'value':df_appa.loc[date_six_kV]['linky'],'index':date_six_kV}

df_appa_10 = df_appa.drop('linky',axis='columns')

for day in jours.values():
    filename = str(day['index'].strftime("%Y-%m-%d"))
    print(filename,day['value'],df_appa_10.loc[day['index']].sum())
    ax = df_appa_10.loc[day['index']].plot.bar(title=str(day['index'].strftime("%Y-%m-%d,
%H:%M:%S"))+' at '+str(round(day['value']))+
'kV',figsize=(10,6)).get_figure().savefig(path+filename+"test.png")
    df_appa.loc[day['index']].to_excel(path+filename+'.xlsx')

def get_df_peak(df,day,minutes=4,threshold=100):
    t = df.loc[day['index']].name.strftime("%Y-%m-%d %H:%M:%S")
    minute = t[-5:-3]
    minute_10 = int(minute)-minutes
    t_10 = t[:-5]+str(minute_10)+t[-3:]
    df_peak = df.loc[pd.Timestamp(t_10):t]
    for appa in df_peak:
        if df_peak[appa].sum()<threshold:
            df_peak =df_peak.drop([appa],axis='columns')
            print('drop ->',appa)
    return df_peak

def get_peak_image(df_peak):
    indexs= list(df_peak.columns)
    dict_peak = {}
    for times in df_peak.index:
        ind_name=times.strftime("%Y-%m-%d %H:%M:%S")+'
P='+str(round(df_appa.loc[times]['linky']))+'kV'
        dict_peak[ind_name]=list(df_peak.loc[times])
    df_peak = pd.DataFrame(dict_peak, index=indexs)
    ax = df_peak.plot.bar(rot=0,figsize=(10,6),fontsize="large")
```

```
    plt.legend(bbox_to_anchor=[0.45, 0.7],fontsize="large")
    plt.ylabel('puissance',fontsize="large")
    plt.savefig(path+"peak"+ind_name.split(' ')[0]+".png")


df_peak=get_df_peak(df=df_appa_10,day=jours[list(jours.keys())[0]],minutes=2,threshol
d=20)
get_peak_image(df_peak)

df_peak=get_df_peak(df=df_appa_10,day=jours[list(jours.keys())[1]],minutes=3,threshol
d=20)
get_peak_image(df_peak)

df_peak=get_df_peak(df=df_appa_10,day=jours[list(jours.keys())[2]],minutes=4,threshol
d=20)
get_peak_image(df_peak)


def get_classifier(df,label_name='linky_binary',max_depth=3):
    features=[]
    for i in range(df.shape[0]):
        ligne = []
        for col_name in df.columns:
            if col_name != label_name:
                ligne.append(df[col_name][i])
            else:
                pass
        features.append(ligne)
    classifier = DecisionTreeRegressor(random_state=0)

#classifier=tree.DecisionTreeClassifier(random_state=0,criterion='entropy',max_depth=
max_depth)
    classifier.fit(features,df[label_name])
    return classifier

def get_feature_importance(df,classifier,label_name='linky_binary'):
    list_importances = classifier.feature_importances_
    print(list_importances)
    i=0
    feature_importance_DT = {'feature':[],'feature_importance':[]}
    for col_name in df.columns:
        if col_name != label_name:
            feature_importance_DT['feature'].append(col_name)
```

```
feature_importance_DT['feature_importance'].append(round(list_importances[i],4))
        i+=1
   return pd.DataFrame.from_dict(feature_importance_DT).set_index('feature')


df_appa_10['linky_binary'] = linky_binary
a=3
t = df_appa_10.loc[six_kV[-a]].name.strftime("%Y-%m-%d %H:%M:%S")
mois = t[5:7]
t_1,t_2 = t[:5]+str(int(mois)-1)+t[7:],t[:5]+str(int(mois)+1)+t[7:]
df_features = df_appa_10.loc[pd.Timestamp(t_1):pd.Timestamp(t_2)]
#df_features=df_features.drop(['freebox','freebox_TV','rasberry_kodi','homeciné','multim
edia','Télé','aspirateur'],axis='columns')
for appa in df_features:
   print(df_features[appa].max(),appa)
   if df_features[appa].max()<1000 and appa!='linky_binary':
      df_features =df_features.drop([appa],axis='columns')


tree_class = get_classifier(df=df_features,label_name='linky_binary',max_depth=3)

feature_importance = get_feature_importance(df_features,tree_class)

feature_importance.plot.bar()
plt.savefig(path+'features_imporantces.png')


day=jours[list(jours.keys())[2]]
t = day['index'].strftime("%Y-%m-%d")
df_day = df_appa_10.loc[t+' 00:00:00':t+' 23:59:00']
```