Machine Learning and Optimization

January 29th 2023

Expe-SmartHouse: Occupancy estimation

Patricia Martínez Ruiz

Leticia Tejedo Cerrato

1. Introduction

The Expe-smarthouse project was initiated in 2018 in order to provide data from a 120 m² household where a 5 people family lives. This project grants access to about 340 measuring points for scientists, accessible in real time through a Grafana portal with Influxdb database.

Figure 1. Smart House.

This smart home is developed based on Open Source Hardware and Software. It is also flexible in adding new technologies or sensors. There are measures of:

- Electricity, gas and water consumption of each device.
- Temperature, humidity and brightness of each common room.
- Opening position of each door and window.
- Motion sensors.
- Light state.
- Air analysis of each room.
- Outdoor weather conditions.

2. Objective

In this project we want to build a model that allows us to determine the presence or not of individuals in the living room of the house. To carry out the estimation, we will need to get access to the different measures from the sensors. For that, the first step is to choose which ones we need.

In the area described we can find the measures of the following figure:

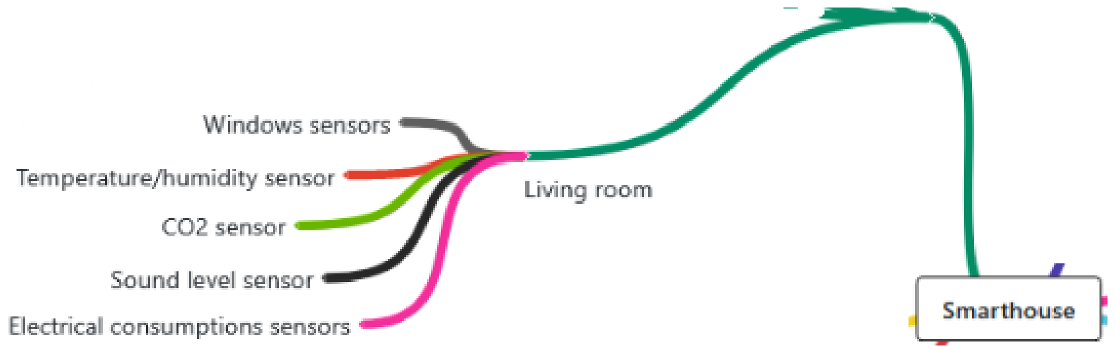


Figure 2. Diagram of the unit studied: Kitchen and living room.

Our goal is to estimate the presence or absence of an occupant. For this we could not use information from outside the house. We therefore need information that is, as much as possible, due to occupancy. For that, we have decided to get access to the following data from the sensors:

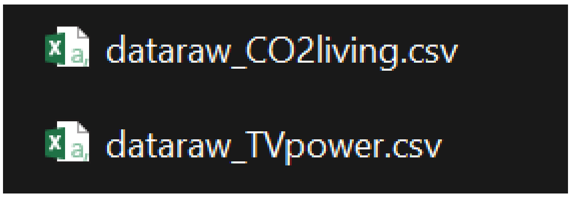
- CO_2 sensor of the living room.
- Power consumption sensor of the TV in the living room.

The objective then is to estimate if there is anybody in the living room of the house, by assuming that if the CO_2 sensor of the living room shows an increase of CO_2, it is because there is somebody in. Same reasoning with the power consumption sensors. If the TV registers any measure, it is because someone has turned on the TV.

3. Data

3.1. Data collecting

Once we have settled up the problem and decided which sensors we need for the estimation, we need to download all the data from Grafana. However, our teacher has provided us with a CSV file for each partially sorted sensor.

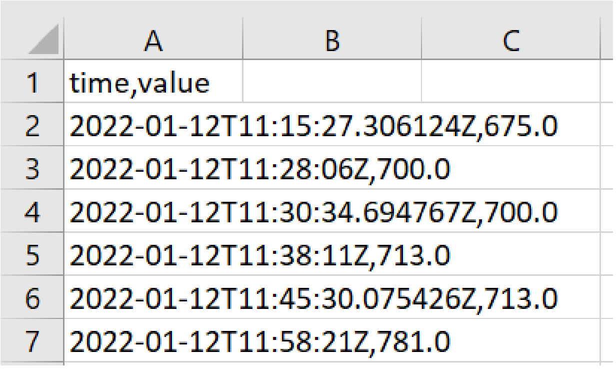


dataraw_CO2living.csv

dataraw_TVpower.csv

Figure 3. CSV file for each partially sorted sensor.

Once this is done, we find ourselves with a raw CSV file with the date and time of each value from the sensors. We need to be careful in this step because we are not ready yet to start exploiting the data. These different CSV files are not usable for our Python code because of the different times of value taking for each sensor. We still have to modify these CSV files before using them to complete our project.



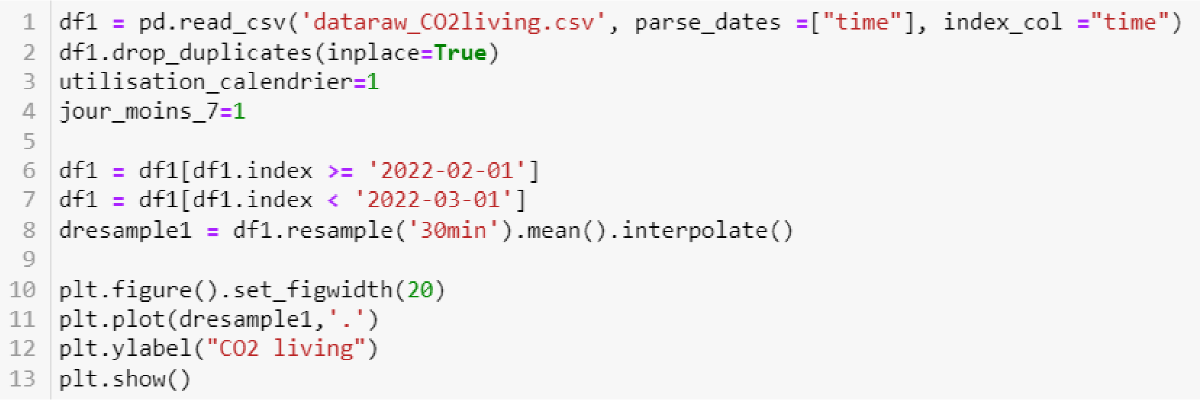
	A	B	C
1	time,value		
2	2022-01-12T11:15:27.306124Z,675.0		
3	2022-01-12T11:28:06Z,700.0		
4	2022-01-12T11:30:34.694767Z,700.0		
5	2022-01-12T11:38:11Z,713.0		
6	2022-01-12T11:45:30.075426Z,713.0		
7	2022-01-12T11:58:21Z,781.0		

Figure 4. CSV file.

3.2. Data treatment

In this first data processing, the different sensors do not have the same period of time, so we have different times and number of data for each type of data. This is very important for the creation of our prediction model because it relates the data at the same time to determine a solution.

Therefore, to solve this problem, we have applied a time step of 30 minutes to the data. In addition, we have set up an interval of time of a whole month, by selecting the date from February 1st to March 1st.



```python
df1 = pd.read_csv('dataraw_CO2living.csv', parse_dates =["time"], index_col ="time")
df1.drop_duplicates(inplace=True)
utilisation_calendrier=1
jour_moins_7=1

df1 = df1[df1.index >= '2022-02-01']
df1 = df1[df1.index < '2022-03-01']
dresample1 = df1.resample('30min').mean().interpolate()

plt.figure().set_figwidth(20)
plt.plot(dresample1,'.')
plt.ylabel("CO2 living")
plt.show()
```

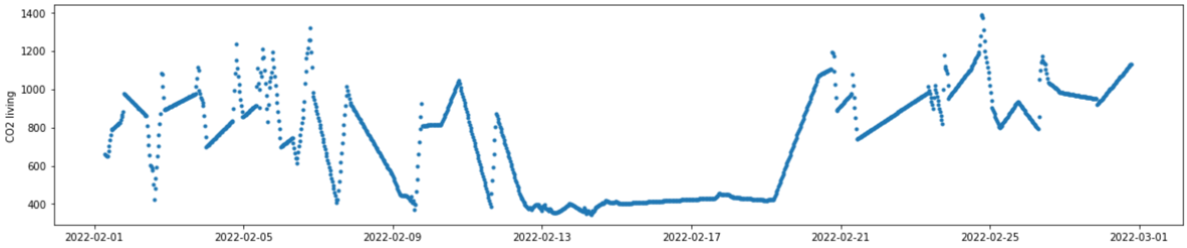
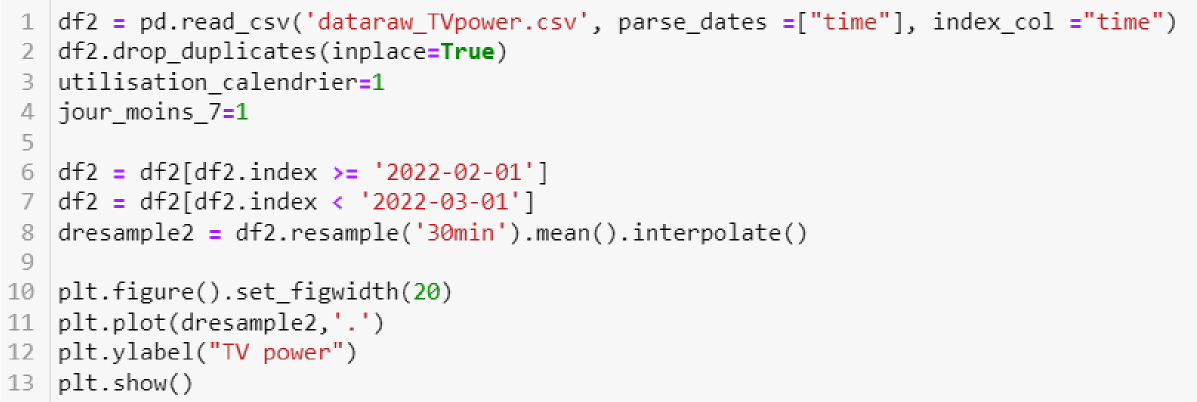


Figure 5. Time step and interval of time for CO2 living room sensor.



```
df2 = pd.read_csv('dataraw_TVpower.csv', parse_dates =["time"], index_col ="time")
df2.drop_duplicates(inplace=True)
utilisation_calendrier=1
jour_moins_7=1

df2 = df2[df2.index >= '2022-02-01']
df2 = df2[df2.index < '2022-03-01']
dresample2 = df2.resample('30min').mean().interpolate()

plt.figure().set_figwidth(20)
plt.plot(dresample2,'.')
plt.ylabel("TV power")
plt.show()
```

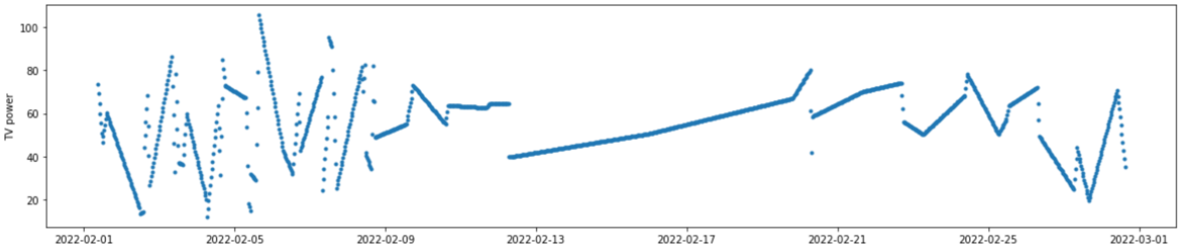
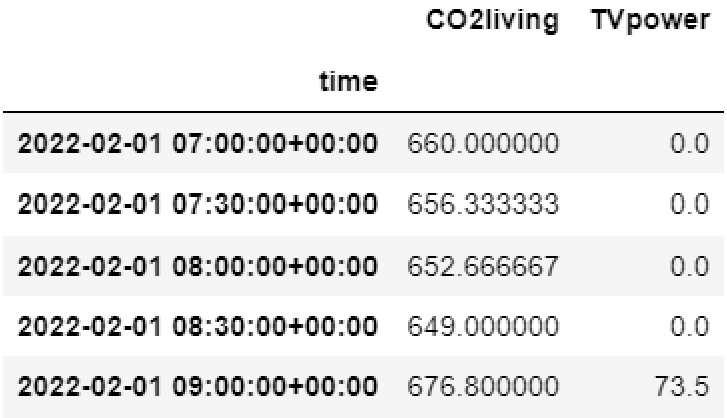


Figure 6. Time step and interval of time for TV power sensor.

With the last code we obtain then all the data splitted in periods of 30 minutes and only for the month of february:



	CO2living	TVpower
time		
2022-02-01 07:00:00+00:00	660.000000	0.0
2022-02-01 07:30:00+00:00	656.333333	0.0
2022-02-01 08:00:00+00:00	652.666667	0.0
2022-02-01 08:30:00+00:00	649.000000	0.0
2022-02-01 09:00:00+00:00	676.800000	73.5

Figure 7. Data frame ready to perform the prediction method.

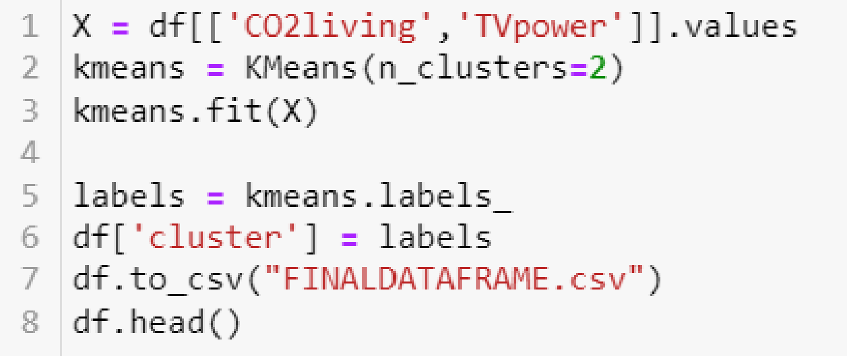
4. Prediction method

As a prediction method, we have decided to use K-Means. K-Means is an unsupervised machine learning algorithm used for clustering. It partitions a set of data points into k clusters, where k is a user-specified number, in a way that minimizes the variance of distances between points in the same cluster. K-Means algorithm iteratively reassigns data points to the closest cluster centroid until convergence.

K-Means can be useful for analyzing sensor data to group similar sensor readings into clusters, allowing us to identify patterns and make meaningful interpretations of the data.

This code is implementing the K-Means algorithm on the dataframe df with two features, "CO2living" and "TVpower" and 2 clusters The K-Means model is fit to the data using the fit method, and the resulting cluster labels are stored in the labels variable.

Then, the cluster labels are added as a new column "cluster" to the dataframe df and the updated dataframe is saved to a csv file named "FINALDATAFRAME.csv".



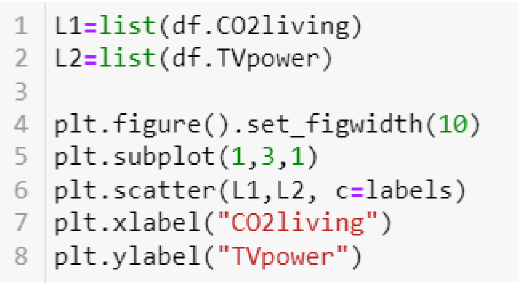
```
X = df[['CO2living','TVpower']].values
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)

labels = kmeans.labels_
df['cluster'] = labels
df.to_csv("FINALDATAFRAME.csv")
df.head()
```

Figure 8. K-means prediction.

5. Results

Finally, we have decided creating two lists, L1 and L2, which contain the values of the "CO2living" and "TVpower" columns of the dataframe df, respectively.

Then, we have created a scatter plot using the plt.scatter method, with L1 on the x-axis and L2 on the y-axis. The c argument is set to labels, so the points will be colored based on the cluster they belong to as determined by the K-Means algorithm.



```
L1=list(df.CO2living)
L2=list(df.TVpower)

plt.figure().set_figwidth(10)
plt.subplot(1,3,1)
plt.scatter(L1,L2, c=labels)
plt.xlabel("CO2living")
plt.ylabel("TVpower")
```

Figure 9. Scatter plotting.

This scatter plot visualizes the data points and their cluster assignments based on the "CO2living" and "TVpower" features. The different colors indicate the different clusters determined by K-Means.

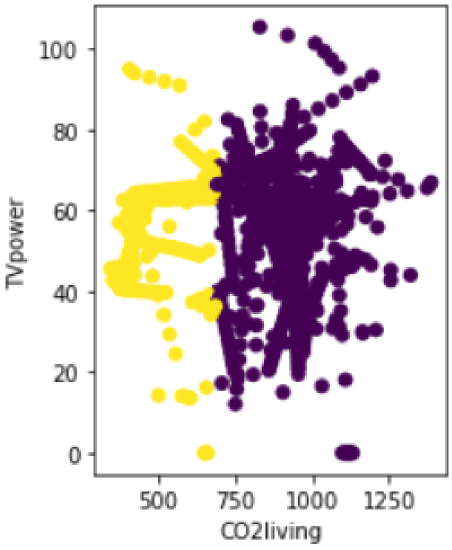


Figure 10. 2 Clusters represented.

In the last graph we can see the 2 clusters represented based on the CO2 values in the room and the consumption of television in it. The purple color shows the cluster for which the CO2 values are high and the energy consumption by the television varies from low to high. We can associate this with the fact that the room can be occupied by people but they cannot have the television on. Also, the yellow color group represents CO2 values that are low, and TV also varies in the same way.

Therefore, we can conclude that the main characteristic is the CO2 in the room, which significantly depends on whether or not there are people in the room. Therefore, the purple cluster would be when there is occupancy, that is, cluster = 1. The yellow cluster would be for occupancy values = 0. That is, there are no people in the room.

6. Conclusion

In this project we have understood and predicted the occupancy of the living room thanks to the smart use of sensors installed in the house. We can conclude that these kinds of smart houses are the future and we must be aware that it is really important to understand the behavior of the occupants in order to improve the home experience and to reach more sustainable ways of living.

But all of this would have been impossible without the amazing machine learning and the artificial intelligence which have given us the opportunity to carry out the estimations and predictions that would have been impossible to carry out by just looking at the data.

Regarding the clustering method, we have been able to identify the most important relationships between occupancy and CO2 levels. The absence of people in the living room means low CO2 emissions. In addition, we have concluded that the fact that the TV shows levels of consumption, it does not mean that there is somebody in the living room because the TV can be turned on but the person can leave the room.

7. Bibliography

- https://miniprojets.net/index.php/2022/07/09/publication-des-bureaux-detudes-gestion-denergie/
- https://miniprojets.net/index.php/2022/02/10/publication-des-bureaux-detudes-smart-systems/
- https://expe-smarthouse.org/index.php/les-capteurs-2/
- https://pypi.org/project/python-sensors/
- https://problemsolvingwithpython.com/11-Python-and-External-Hardware/11.04-Reading-a-Sensor-with-Python/