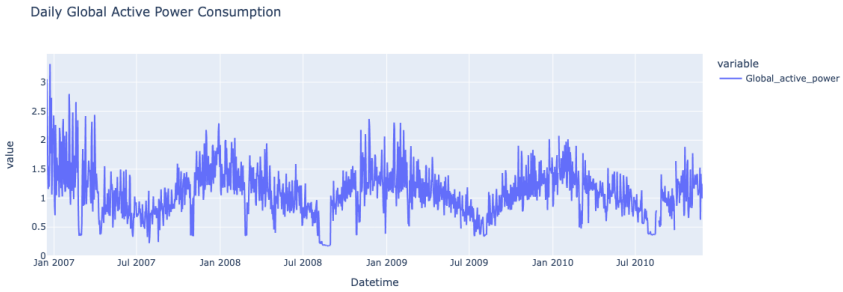
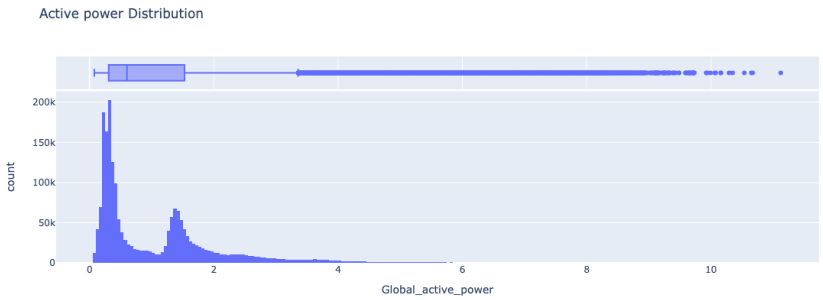
PRUDHOMME Pierre

Electric energy comsuption

We are working on a electrical comsuption dataset in one French household with a one-minute sampling rate on 47 months. We have over 2 millions observations and 9 features to try to understand our dataset. We have the '*Global_active_power*' which will be our target for this analysis, it represents the household global active power every minute and it's in kilowatt. We also have variables giving us the comsuption of 3 parts of the house in the variables : '*sub_metering_1*', '*sub_metering_2*' and '*sub_metering_3*'. We have voltage, intensity and reactive power left in features.
We can start our analysis by drawing some plots to help us understand the dataset we are working with :

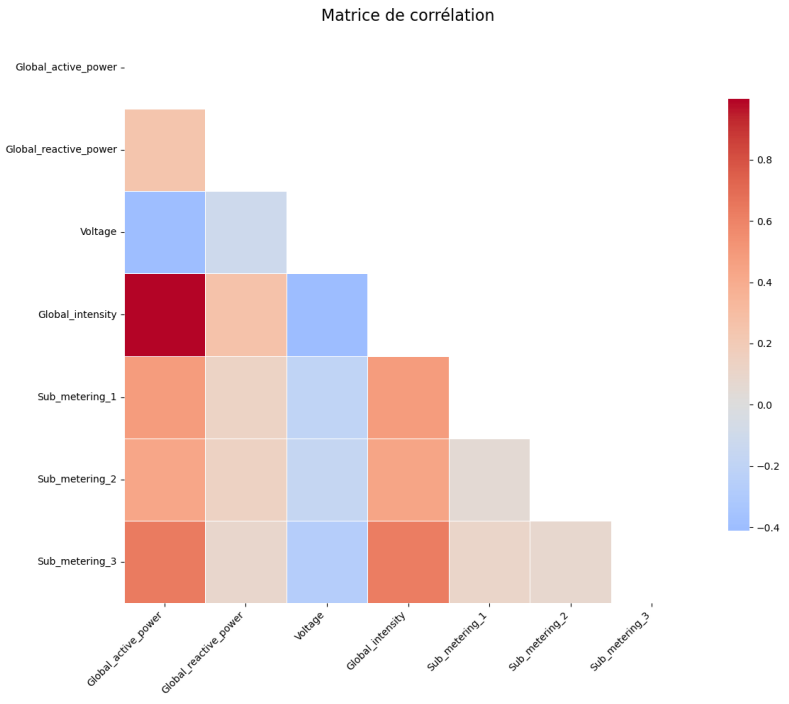
- The first plot is the values of '*Global_active_power*' mean by day. We see that the comsuption is higher in winter every year and in the summer it tends to reduce. They are also pick during each week representing week-ends or when people are at the house the whole day or even sometimes outliers





- In the second plot we can look at the distribution of the active power and see that it highlights 2 bumps. Those bumps represents the different comsuption, when high demanding machines are working (oven, electric water-heater, …) and when just classic electrical stuff is working.

We need to understand how of the active power is link to other variables and what remarks can we make of those correlation. So we build the correlation matrix :



We can see that our target has the correlation of 1 with Intensity, that's explained by the formula :

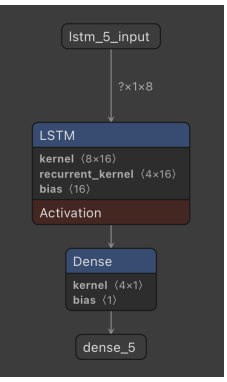
P = U * I

So in order to prevent a problem, we need to take care of that variable.

We also see that our target is explained a lot by the '*sub_metering_3*' variable, this variable is the electric water-heater comsuption, the biggest electric consumption in the house that's why it's the most correlated variable. Other sub_metering variables represents the kitchen and the laundry room, two very energy-consuming rooms.

In this dataset we had also missing values that we needed to take care of. Those values were a day missing or even two consecutive day missing, surely a problem coming from the acquisition. We choose to replace those values with the *mean* of each columns to not create outliers values. We could have also replaced our missing values with median or maybe a normal distribution.

We try to build our LSTM model to predict the '*Global_active_power*' value. We reach a very small loss of 10e-3 as seen on the model loss figure. That seems like we have an overfitting issue… That means our model is too much accurate to be predicting real values, so we need to get the most correlated variable out and build a new target.



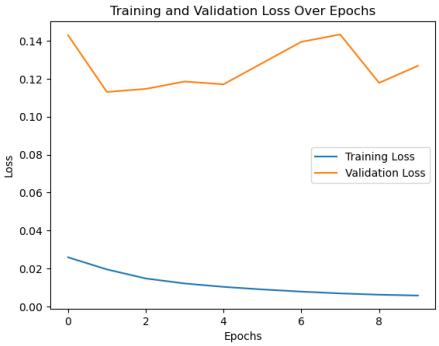
This is architecture of our first LSTM model, the architecture consists of an LSTM layer with 4 units that processes time sequences, followed by a Dense layer with a single output to predict a continuous value, such as '*Global_active_power*' or '*active_energy*'This enables the model to learn temporal relationships in the data and provide predictions based on those learnings.

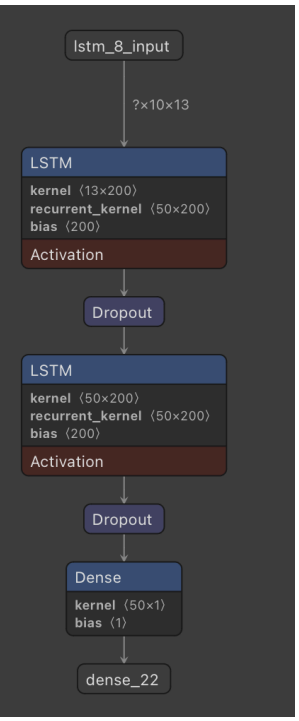
We build now the active energy variable with all the others measurements with the formula :

(global_active_power*1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3) = active_energy

After creating this new variable, we get rid of the variables used for the calculation and we print a new correlation matrix and see that there I no highly correlated variable that can biased our model. And we have a new target that will reduce the calculation time of the biggest model in order to get a good result with less calcul time. My method here can be to predict '*Global_active_power*' also.

We also go the same result with our CNN 1D model, a loss around 10e-3. We need to look at the validation loss in order to see if there is really and overtfitting problem or if the model is just really performing. The validation loss came around 0.14 for the 2 architecture. It is a problem that means our model is having hard time trying to generalize the data.





Then I tried a more complex architecture on the LSTM in order to obtain a smaller validation loss to create a model which is capable of more generalization. This model consists of two LSTM layers, with 50 units each, designed to capture temporal patterns in sequential data, with the first LSTM layer returning the full sequence and the second reducing it to a single output. Dropout layers with a 20% dropout rate are included to prevent overfitting, and a final dense layer outputs the prediction.

And this one gave to me very good results, the validation loss and the loss are on the same level so this model works better for generalization but it takes more time to compile. Here we train it with 10 epochs and a big batch size of 128.