POLITECNICO
MILANO 1863

Financial Engineering

Assignment 1 EPLF, Group 16

2023-2024

Alice Vailati - CP: 10683600 - MAT: 222944
Andrea Tarditi - CP: 10728388 - MAT: 251722
Jacopo Stringara - CP: 10687726 - MAT: 222456
Nicolò Toia - CP: 10628899 - MAT: 247208

Contents

1. Introduction and data

In this report we explore applications of the LEAR model with various regularization techniques to minimize the weights of our model.
To begin our exploration we took a look at the provided dataset. We have about 26k observations of various energy market quantities. To be more precise the provided quantities are:

- *TARG__EM_price*: the price we will try to forecast.
- *FUTU__EM_load_f*: the forecast load.
- *FUTU__EM_solar_f*: the forecast solar radiation.
- *FUTU__EM_wind_f*: the forecast wind speed.
- *CONST__wd_sin*: the sine of the weekly seasonality.
- *CONST__wd_cos*: the cosine of the weekly seasonality.
- *CONST__mnth_sin*: the sine of the monthly seasonality.
- *CONST__mnth_cos*: the cosine of the monthly seasonality.
- *CONST__yd_sin*: the sine of the yearly seasonality.
- *CONST__yd_cos*: the cosine of the yearly seasonality.
- *Hour*: the hour of day.
- *Date*: the date.

Before running the experiments we expect the LEAR model, choosing the "optimal" Lasso to outperform the canonical ARX with fewer significant weights.
Going in, since we are forecasting energy prices we expected the previous day's prices to have a greater impact than all other features. We also expected the load to have a noticeable impact throughout the day, especially during peak hours. Regarding the weather forecasts, our expectation was that solar would be more relevant during the middle of the day (i.e. around 12 AM) while we expected the wind speed influence to be more scattered during the day.
As a final remark, we expected seasonality to be moderately influential since energy consumption is well known to be related to the seasonality, especially of the month and weekday.

1.1. Experiment Methodology and Criteria

Our approach consisted in training the model on the same dataset and same dates, with different values of the regularization parameters λ_1 and λ_2.
Then we analyzed the weights and error functions, in order to understand whether some patterns and trends emerged in the model and whether it was actually learning and generalizing the training data or simply learning from noise.

2. Experiments

Initially we selected an ARX model. Hence our prediction can be formulated as follows:

$$\hat{y}_{t+h} = f_\Omega(y_{t-k:t}, z_{t-k:t}, x_{t+h})$$

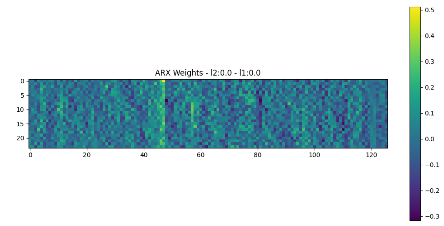
where h is the prediction horizon (in our case the next 24 hours) and k is the maximum ARX lag (in our case the last two days).
In the case of the LEAR model our forecasting function f_Ω is a simple affine transformation from the feature set and past values set to our target values. Let us also remark that in our case we are using 126 features made up as follows:

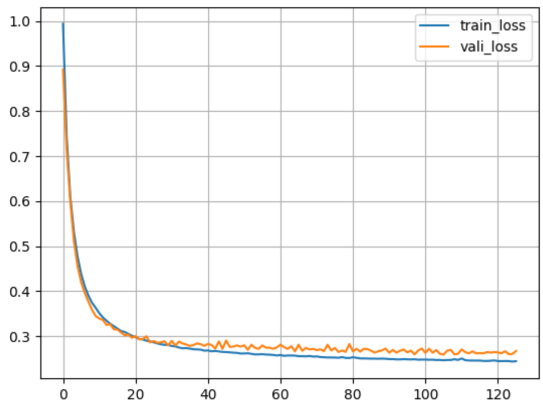
- The first 48 features are the past prices from the last 2 days (24 x 2)
- the following 72 features are the forecast values for load, solar and wind for the day (24 x 3)
- the last 6 features are the seasonality sine and cosine for the weekday, month and year

To have a benchmark against which to evaluate our future explorations we first run the LEAR without any regularization (i.e. by setting $\lambda = 0$). In our case we chose to use the MAE loss function:

$$\min_\Omega \sum_{n=1}^{N} |f_\Omega(x_n) - y_n|$$

Training this model yields the following weights and losses for the first day:





In the following we will always analyze the weights and losses for the first day since we noticed that the performance were rather similar across the 10 days we took into consideration.
As we can see the weights are all quite diverse and different from zero either positive or negative. This was to be expected, since there is no regularization, the model is not encouraged to diminish the size of the weights and hence "select" the features to use. Furthermore, a clear pattern in the weights cannot be made out, other than a slight preference for prices of the previous day's last hours.
Indeed, if we take a look at the loss on the training and validation set we can see that the model start to have a higher loss on the validation set than on the training set after about 20 iterations. This means our model is starting to learn from the noise contained in the training data, in other words, we are over-fitting to the training data.
The model produces the following predictions:

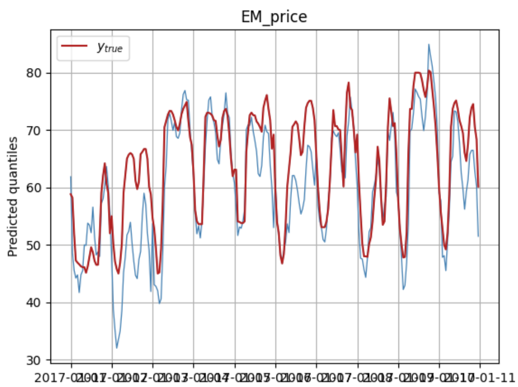


Figure 1: Energy prices predictions

As we can see, while the performance of the model is relatively good there are some peaks in which our prediction our greatly off the mark.

2.1. LASSO regression

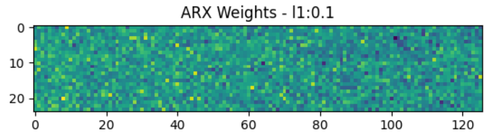
To improve our results we explored our first regularization technique: LASSO regression. This technique consists in modifying the function to be minimized as follows:

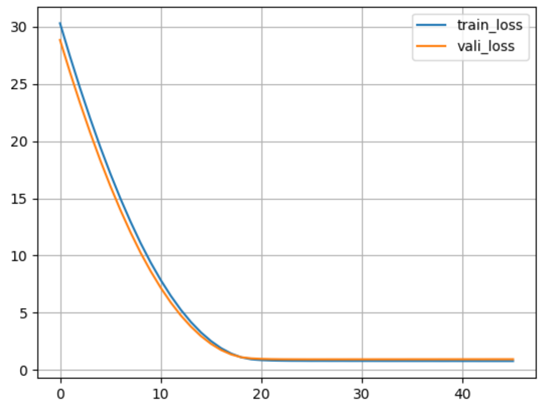
$$\min_{\Omega} \sum_{n=1}^{N} |f_{\Omega}(x_n) - y_n| + \lambda_1 \cdot \sum_{j=1}^{\#\Omega} |\omega_j|$$

This new second term, as we can easily see, incorporates the estimated weight of our ARX model into the loss function. In other words, we add a further object to our original function, no only to minimize our error with respect to the target value but to also minimize the weights with which we achieve this estimation.
In order to explore different values of λ_1 we plotted the weights and loss functions for different values of λ_1.

2.1.1 Lasso with 0.1

We started with a more drastic value for λ_1 of 0.1. Let us take a look at the weights and loss function:





As we can see the cross between the train and validation loss happens even earlier than in the base case. This is probably due to the fact that such a drastic value of the regularization parameter does not allow the model to properly learn and generalize the training data to the validation set.
Furthermore, the weights in the heat map are scattered but all very close to zero, around two orders of magnitude less than in the base case.
Indeed if we also take a look at the predicted data we obtain the following:

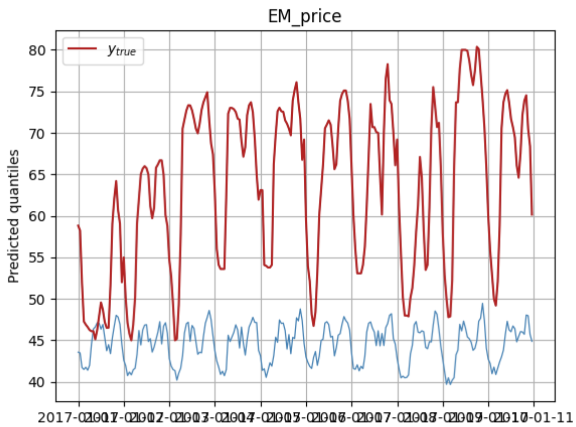
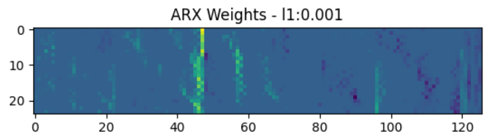


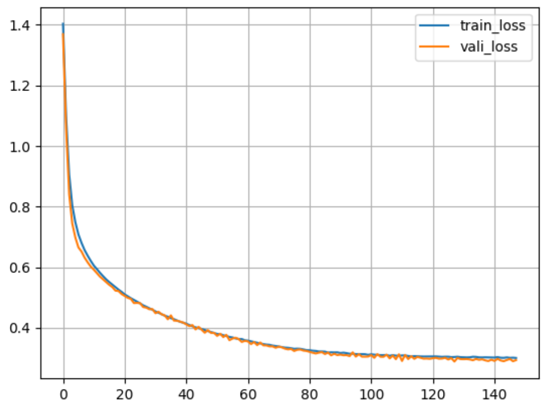
Figure 2: Predicted data with $\lambda_1 = 0.1$

As we can see the weights are all pushed so much towards zero that the predicted values are extremely far from the true values. Probably, such a high value of λ_1 is pushing all weights to be of a full order of magnitude less than they should be even if the features are actually significant to the model. Hence this leads to the weights being more comparable in terms of absolute value with each other and thus not achieving the true objective of a LASSO regularization.

2.1.2 Lasso with 0.001

Now we evaluate our model with a lower value for λ_1 that is less drastic and hence relaxing our constraints on the weights. Let us analyze the plot of the weights and train and validation errors:





As can be seen from the left-hand plot, most weights are still being pushed towards zero while some manage to be much greater than zero. In particular we noticed a few interesting possible patterns. The most influential features are by far the prices of the last hours of the previous day, hence we can see a strip of meaningful weights around 45-47, this is probably due to the fact that prices are, on a macroscopic lens, ”continuous” processes and thus present some inertia.
The load also presents some relatively significant weights especially around the 10 AM (about 56 on the x-axis) mark and a softer influence around 6 PM (about 65). In our opinion this can be explained by the fact that the morning’s demand sets a reference level for the rest of the day while 6 PM is a natural peak associated with people coming back home.
Solar radiation, contrary to what we expected, does not present any significant weights. Even during the central hours of the day, they do not influence the price much. Perhaps, this could be explained away using the fact that our forecast period is of 10 days in the middle of January (winter).
On the other hand, the wind presents two interesting patterns. The first presents positive weights relating to the early hours of the day (the peak around 96 on the x-axis). The second is a negative trend throughout the rest of the day. Our explanation for these phenomena has to do with daily wind patterns. Winds are usually slower during the first hours of the day while they tend to pickup as night approaches.
We can also see that the weekday seasonality has a slight positive influence.
To support these interpretations we can also notice that the right-hand graph does not show a clear cut-off point in which our model clearly deviates from the validation data. We interpreted this to mean that our model fitted the data well but that it picked up pattern and generalized them.

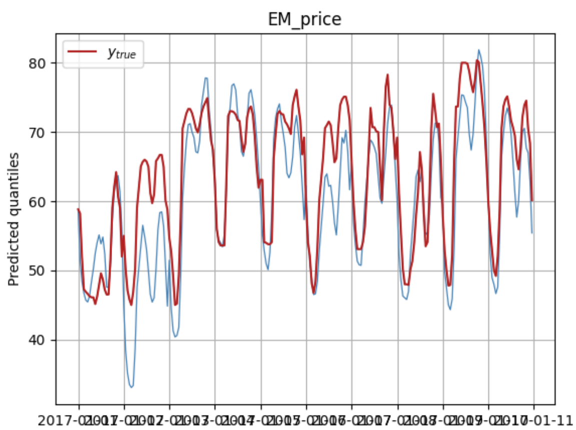
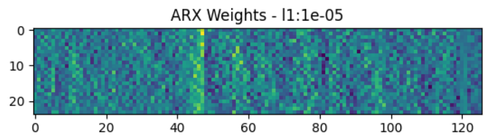


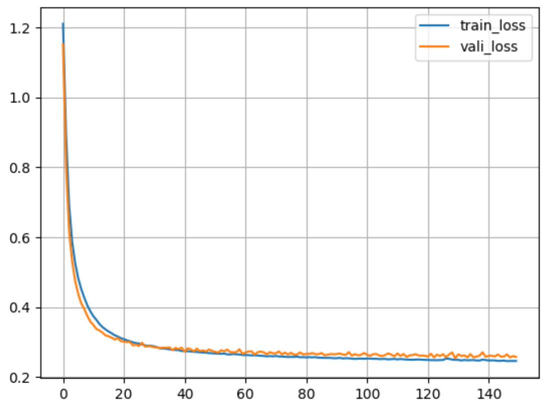
Figure 3: Predicted data with $\lambda_1 = 0.001$

We also noted that the forecasting is slightly more accurate on the test set. While we are still not capturing the full process our error around the peaks are much less prominent.

2.1.3 Lasso with 10^{-5}

Now we try to further relax by two orders of magnitude the previous λ_1 in order to understand how strong of a regularization we actually need.





First of all, we analysed the weights plot. We can easily see that here we have a distribution quite similar to the base case rather than to the case with $\lambda_1 = 0.001$.
Indeed, if we also take a look at the right-hand graph, we can notice that, just like in the base case, the model tends to over fit the training data rather early.

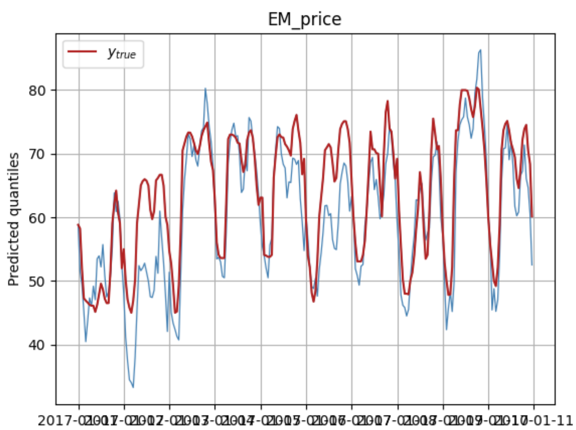


Figure 4: Predicted data with $\lambda_1 = 0.00001$

No marked improvement can be noticed from the graph of the predicted quantity against the true values.

2.2. Ridge regression

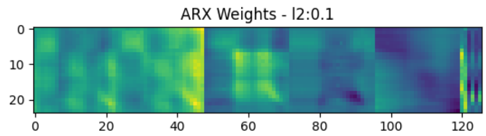
The Ridge regression is another regularization technique: in this case the penalisation added to the function that has to be minimized is the sum of the square of the weights. In other words, while the LASSO technique regularizes the weights using the ℓ_1 metric here we use the ℓ_2 metric. Hence the minimization problem becomes:

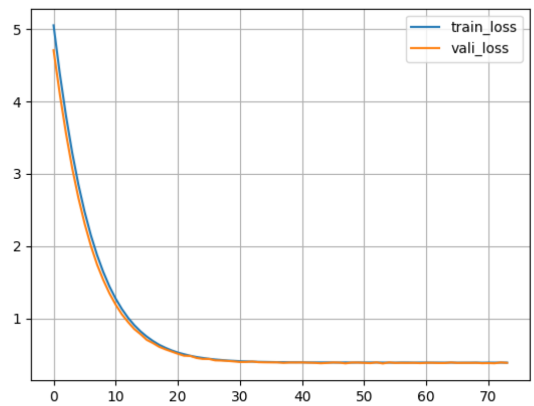
$$\min_{\Omega} \sum_{n=1}^{N} |f_\Omega(x_n) - y_n| + \lambda_2 \cdot \sum_{j=1}^{\#\Omega} \omega_j^2$$

Usually, while LASSO regression is used to perform a sort of "feature selection", Ridge regression is used to avoid over-fitting, which as we have seen has been a problem for our model in the previous section.
In order to better understand if this method leads to some kind of improvement to the forecasting, we decided to use the first two values used in the previous part but applied to λ_2.

2.2.1 Ridge with 0.1

As with the LASSO case we started out with a rather drastic value of $\lambda_2 = 0.1$.





If we look at the left-hand plot we can notice, that similarly to the case of $\lambda_1 = 0.1$ all weights are being pushed towards zero. However, some patterns are more recognizable.
If we look at the right-hand graph, we can see that the Ridge's main objective is reached since there is no clear over-fitting. On the other hand the loss remain overall quite high.
Indeed if we take a look at prediction:

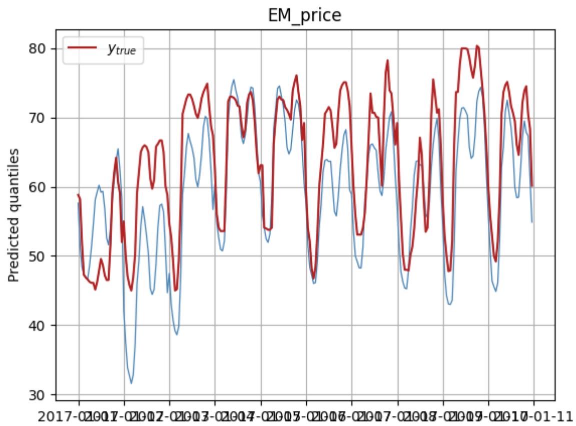
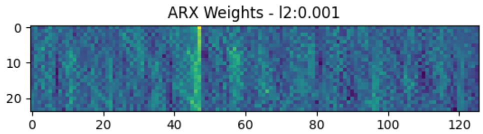


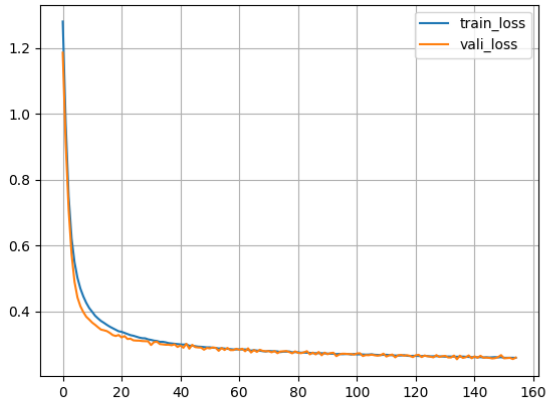
Figure 5: Predicted data with $\lambda_2 = 0.1$

As expected from the error plot we can notice that while there is no clear over-fitting the model fails to correctly predict the prices in a few key points.
In particular we seem to underestimating the values. This underestimation may be caused by a high value of the λ_2 which pushes the weights towards zero.

2.2.2 Ridge with 0.001

Now, just like we did for the LASSO regression, let us try a more relaxed value for the regularization parameter $\lambda_2 = 0.01$.





If we look at the error, we can again see that no major over-fitting is present. A pattern also seems to appear in the weights distribution even though all the weights overall seem to be quite scattered. Overall the model, while flawed, is in our opinion still valid.

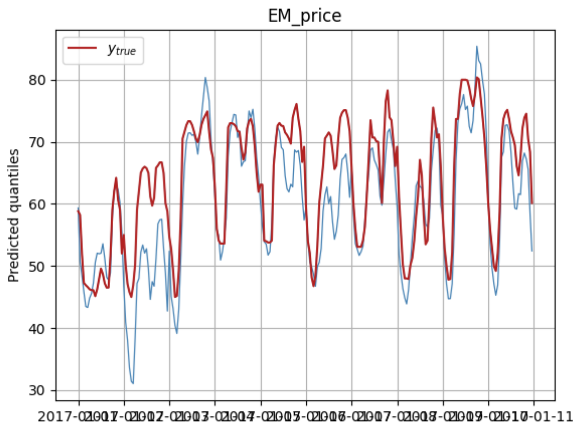
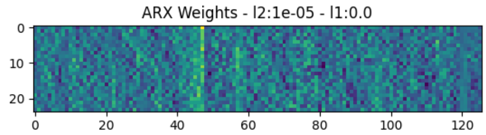


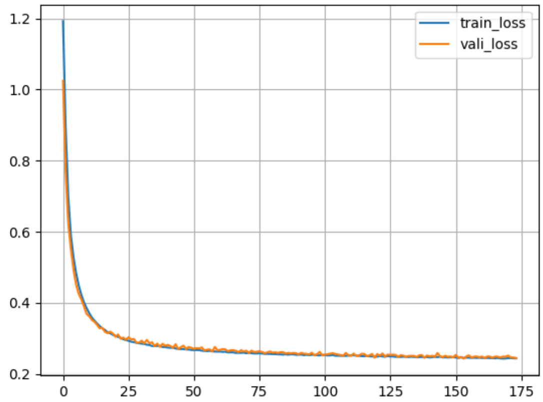
Figure 6: Predicted data with $\lambda_2 = 0.001$

If we instead take a look at the prediction we can see a marked improvement in the later days while we can also observe a severe mispricing in the earlier period.

2.2.3 Ridge with 10^{-5}

Finally we also performed the Ridge regression with a rather relaxed value of $\lambda_2 = 10^{-5}$. Let us take a look at the weights and errors:





Again, no severe over-fitting is present but the weights are rather sparse in a similar way to the base case and no clear-cut pattern can be noticed.

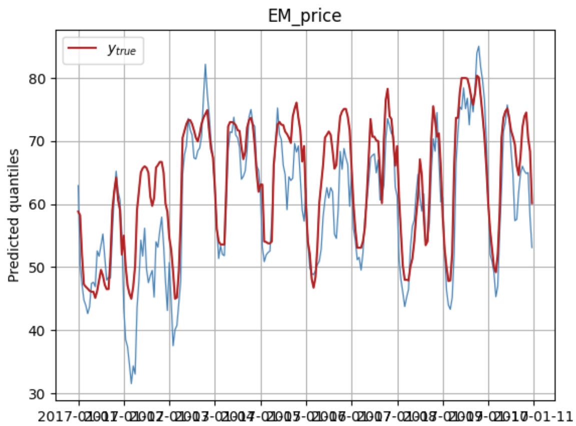


Figure 7: Predicted data with $\lambda_2 = 10^{-5}$

Similarly to the case of $\lambda_1 = 10^{-5}$, such a small regularization parameter does not influence the model in a significant way. The graph does not present a noticeable improvement from the base case.

2.3. Elastic Net regression

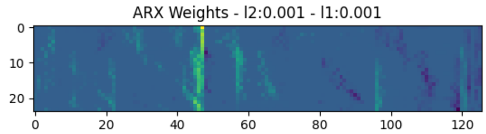
Lastly to find a more balanced middle-ground we performed an elastic net regression which merges the two regularization techniques. The objective function becomes:

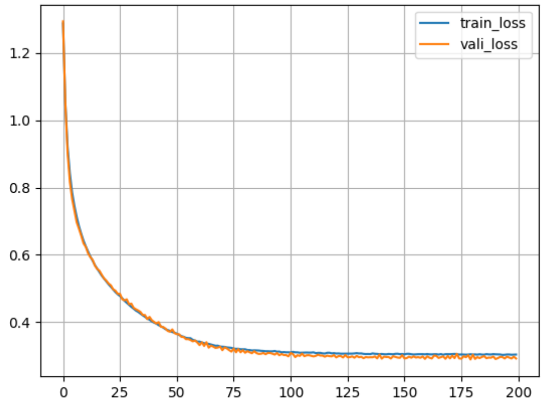
$$\min_{\Omega} \sum_{n=1}^{N} |f_{\Omega}(x_n) - y_n| + \lambda_1 \cdot \sum_{j=1}^{\#\Omega} |\omega_j| + \lambda_2 \cdot \sum_{j=1}^{\#\Omega} \omega_j^2$$

This techniques offers the best of both LASSO and Ridge. It should perform a ”feature selection” like LASSO and avoid over fitting like Ridge.
We decided to investigate the case with values of $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$ or $\lambda_2 = 10^{-5}$. We chose these values since $\lambda_1 = 0.001$ provided by far the best results for the LASSO regression while the two values for λ_2 performed both rather well.

2.3.1 Elastic Net with 0.001 and 0.001

First off we performed an Elastic Net regression using $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$.





We can clearly recognize the same patterns that emerged with the case of Lasso $\lambda_1 = 0.001$ but perhaps even more clearly thanks to the influence of λ_2. The loss seems to decline a bit sharper than in the LASSO case and to reach a lower final value.

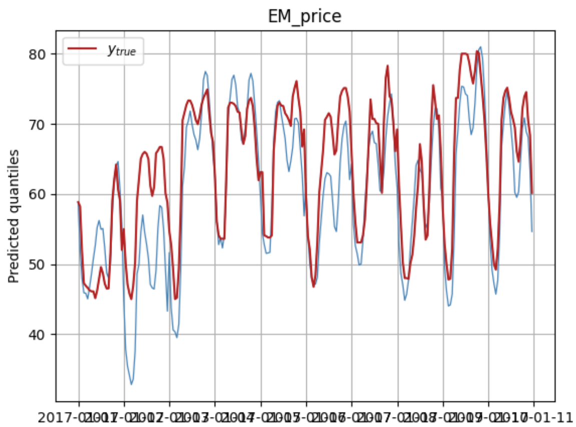
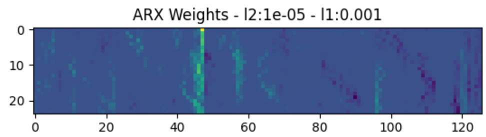


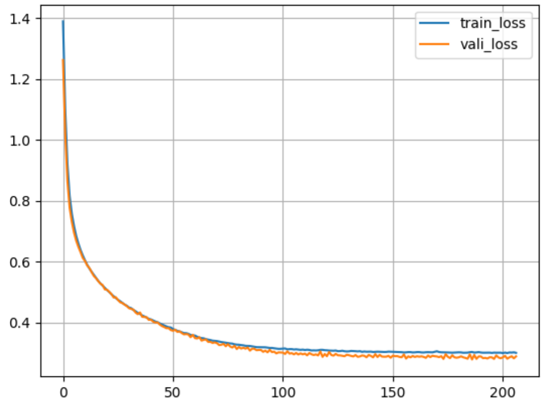
Figure 8: Predicted data with $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$

The performance on the test set seem still rather good even though some mistaken peaks are still present.

2.3.2 Elastic Net with 0.001 and 10^{-5}

Now we pass to relax the Ridge regression a bit further thus we performed an Elastic Net regression using $\lambda_1 = 0.01$ and $\lambda_2 = 10^{-5}$.





Again, the weights plot is rather similar to the previous case, and some weights are pushed a little bit more towards zero. While the error plot still does not present over-fitting although the final error seems to lower further.

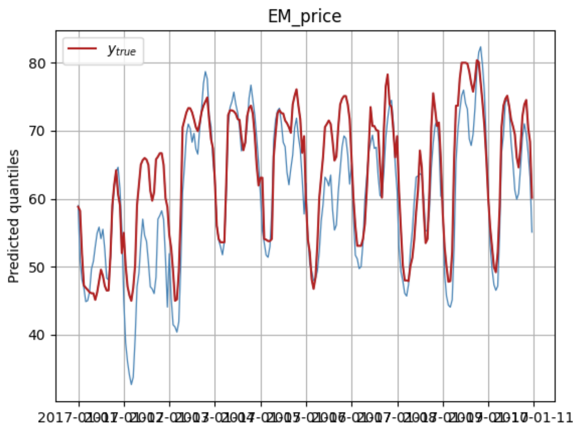


Figure 9: Predicted data with $\lambda_1 = 0.001$ and $\lambda_2 = 10^{-5}$

The predicted values are still quite similar to the previous case and the mistake peaks are still present. This was to be expected, relaxing the overfitting regularization parameter does not necessarily improve the performance on the test set.

3. Metrics

After experimenting with various different combinations of λ_1 and λ_2 we decided to look for the better model by evaluating a variety of metrics on the test set predictions.
For each model of interest we evaluated the RMSE, MAE and sMAPE as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

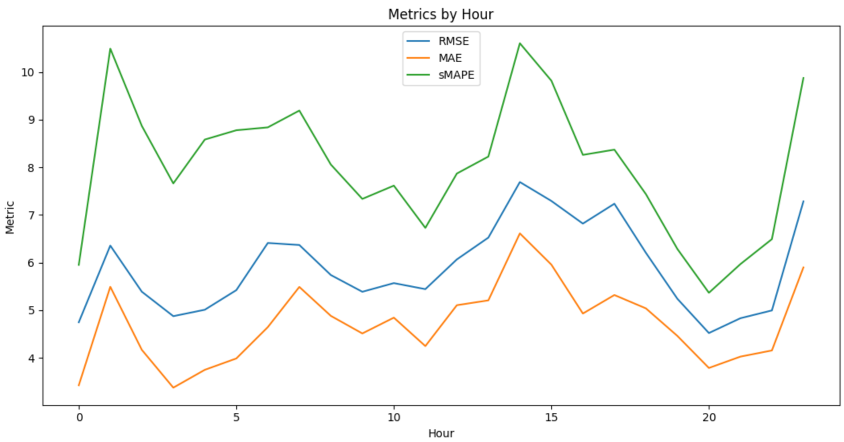
$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

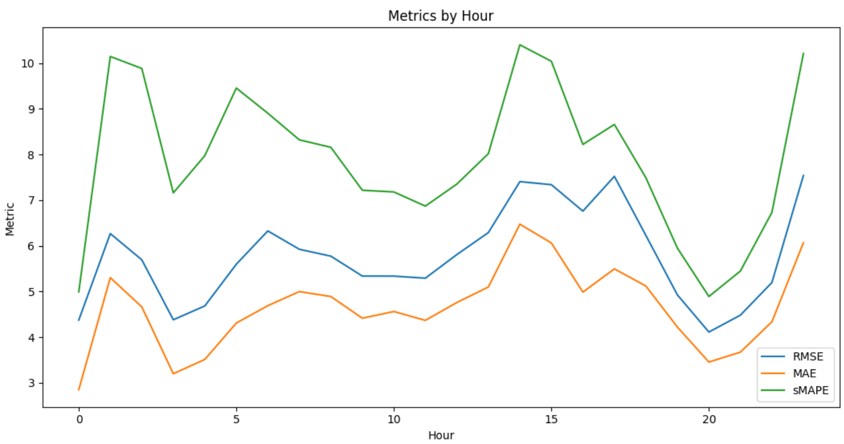
$$\text{sMAPE} = \frac{2}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

	Base Case	Lasso 10^{-3}	Ridge 10^{-3}	EN 10^{-3} , 10^{-3}	EN 10^{-3} , 10^{-5}
RMSE	6.23	5.89	6.24	5.99	6.02
MAE	4.90	4.63	4.84	4.79	4.69
sMAPE	8.28%	7.87%	8.25%	8.16%	8.03%

Table 1: Metric for different model choices

We can see that most models all perform similarly. The only two models that really outperform the others in terms of all metrics on the test set are LASSO with $\lambda_1 = 0.001$ and Elastic Net with $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$. Furthermore also the Elastic Net $\lambda_1 = 0.001$ and $\lambda_2 = 10^{-5}$ performs rather well, especially in terms of the MAE. Let us also note that the base case gets outperformed on the test set by almost all regularized models regardless of the specific technique applied.
We also found it interesting to note how the various metrics vary in relation to the regularizing parameters variations.
The lowest MAE, which is a type ℓ_1 metric, achieves its lowest value with the ℓ_1 regularization of the LASSO technique. On the other hand the RMSE, which is a type of ℓ_2 metric, does not follow the same pattern. Perhaps this could be explained by the form of the chosen ARX model. In other words, a model with these many features benefits much more by a feature selecting technique such as LASSO than from a Ridge regression. Thus our two candidates for best model were the LASSO $\lambda_1 = 0.001$ and Elastic Net $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$. Therefore we chose to compute and plot their metrics grouped by hour of the day:



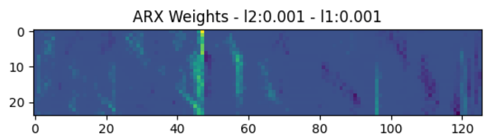


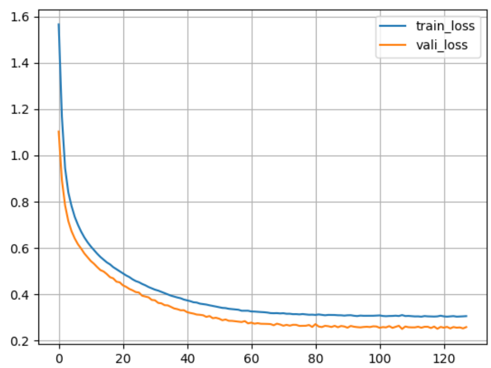
Here the first graph refers to the LASSO model while the second to the Elastic Net model. As we saw above both methods yield rather similar errors and as can be seen from the two plots they present relatively similar errors throughout the day's duration.
However, it can be easily appreciated that the EN model has in general a smoother error function. Indeed it does not present the same error peaks as the simpler LASSO regression.

In other words, this means that while the EN model has overall a higher error, this error seems to be more consistent across all hours of the day.
Thus, we chose to use the EN model to obtain a more general and consistent result even though the overall error is a bit higher.
Analyzing more in detail the behaviour of the error, both peaks and valleys can be explained by a qualitative approach. The valley around 20 pm can be explained by the fact that in this time slot the demand, and hence the price, is more predictable. On the other hand, during the night the price drops from its highest point at around 20 pm to its lowest around 6 AM, hence our model struggles to capture such a steep decline. This results in a non-negligible error. The last peak around 2 PM is probably related to the unpredictability of each day's weather condition that is then related to demand and price.

4. Seasonality

As a final experiment we also wanted to explore the role of yearly seasons on our model. In order to do this we moved the test window from January to July, keeping a 10 day period.
Calibrating an Elastic Net model with $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$ yields the following plots for the weights and error:



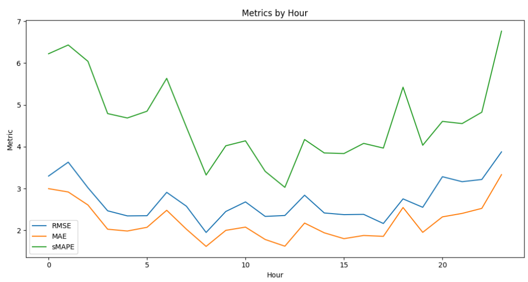


No notable difference in the weights can be spotted. This could be explained by the fact that the model, being trained on the whole year, already generalizes what it can and deals with the different season conditions without altering the weights too much.
Indeed, it yield the following metrics:

RMSE	MAE	sMAPE
2.75	2.21	4.65%

Table 2: Model metric in summer

which by hour perform as follows:



The fact that the hour error seems to have an inverse trend with respect to the winter scenario can probably be attributed to different habits of the population due to the season and climate.

Finally, the great decrease in the overall error can simply be attributed to the diminished price of electricity during the summer. Indeed if we take a look at the predicted data and the real data we can see that the energy price is much lower than in the winter scenario:

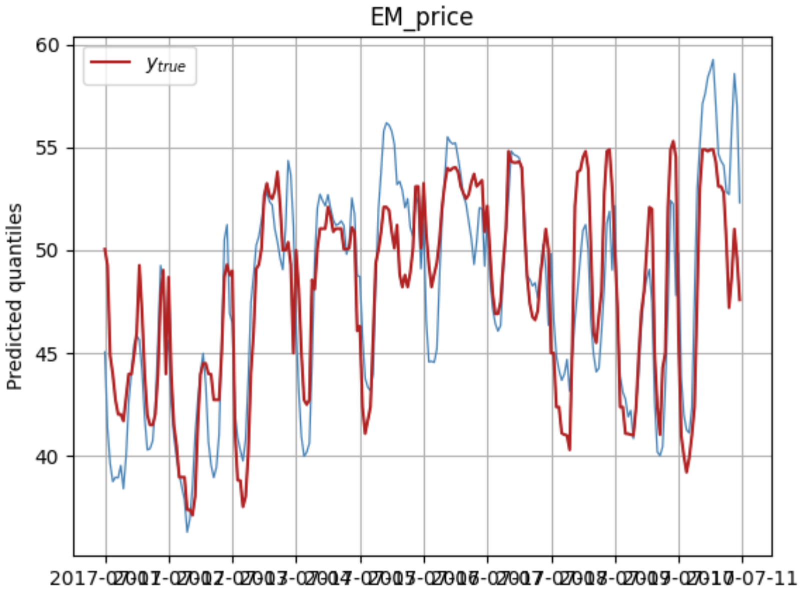


Figure 10: Predicted price in the summer

5. Reference

- Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, J. Lago, G. Marcjasz, B. De Schutter, R. Weron