X/HEC TIME SERIES 2024 - Practical Session Report – AirIsGood Team

Group Members:
Carlo Antonio PATTI, Antonio Roberto VENTURA, Ghali CHRAIBI, Jules CREVOLA

GitHub repository with every notebook:
https://github.com/julescrevola/air_quality

Data Preprocessing

We included many different steps to our data engineering and preprocessing.
First, we found some Meteo France data online for past weather and combined this data with our training set, keeping only relevant columns. We believed that weather could indeed impact a lot the concentration of gas molecules in the air, for example the heat favoring a higher concentration.
We then added a dataset of traffic flow and occupation rate in Paris during the period. It was challenging to set up as we had to combine many text files together, and a csv file on top of it, to cover the whole period of the train set but we were convinced that it made a lot of sense, as more traffic means more pollution usually. Regarding traffic, we also added a feature for rush hours, in the morning and late afternoon, as to put more weight on periods where the air contamination was supposedly higher due greater traffic.
Then, as the train set covered the Covid-19 period, with the lockdowns and curfews, we also inserted categorical variables for these aspects, as it meant less activity during these periods and thus probably less pollution.
Finally, we added a dataset of French holidays to our train set, as we believed that it could have an impact on pollution too (during Parisian holidays, people would maybe leave the city, while during others regions' holidays people may come to the city, influencing on the levels of activity and thus gas concentrations in Paris)/

For each feature added from these additional datasets, we plotted the kernel density of the target variables to understand the impact (see example for rush hours in Fig1.).

For the dates, we created features for the year, the month, the day, the hour, the day in the week and whether it was a week-end or not as to gather as much information for each time step as possible.

We added three lags as features: one 1 hour before, one 12 hours before and one 24 hours before to better capture the trends of molecules' concentrations.
Finally, we used linear interpolation to fill in for missing values in the train set.

We then proceeded to preprocessing our variables. We used a MinMaxScaler for numerical variables, which divided the difference between the actual value and the minimum value by the difference between the maximum and minimum values. We used a OneHotEncoder for categorical variables to give weight to each value of each category, and finally used Cyclical encoding for the date features as to represent better the evolution of time in our train set.

We divided the train set into 5 different ones, as we wanted to train one model per target variable and then combine the predictions.

Model Architecture and Training

We tried various models. First, we used a simple linear regression to set a baseline, and then went on more complex models. We ran a loop of around 10 regressors including XGBoost, CatBoostRegressor, etc (see Fig2.) and ranked them according to their mean MAE score and MAE standard deviation. We ran a model using the DARTS method, but the result was not very interesting. We also tried to run a neural network with LSTM layers and GRU layers, but the results were less promising and the MAE was quite big, which led us to focus more on more classical machine learning models.

Looking at the results of the loop running over the various regressors, we decided to model our predictions with a CatBoost regressor as it had the lowest mean MAE score and a very reasonable MAE score standard deviation. It made sense as the CatBoost regressor is very good at managing categorical columns and they were the majority of our features. Once we chose the CatBoostRegressor model, we tuned its hyper-parameters for each target, using TimeSeriesSplit as the cross-validation method to fit better the time series behaviour.

Model Performance and Comparison:

- Aside from the various regressors that we ran, we tried an approach with the ARIMA method, but the results were very bad and the predictions did not follow at all the trend of the targets. The implementation was maybe wrong but we did not have time to focus on this one and discussing with other teams we came to the conclusion that it was not worth it spending too much time on that.
- The deep learning model that we ran, one with LSTM layers and on with GRU layers, were promising at the end but the final MAE was quite high (around 13) and they were also difficult to understand and optimize, which made us explore other ways with which we were more comfortable and whose output we could understand better.
- Finally, we landed our choice on the CatBoost regressor, as explained above, and it gave us a MAE of around 9 on our test set. It was not very good but still better than the rest so we kept it and tried to optimize it with hyper-parameter tuning specific to each target.

Discussion

Overall, the work conducted for data engineering and preprocessing, with the research of external datasets that could complete the train set, the reflection on how to best use the features that we could get and how to preprocess our data, especially for dates, was a great a learning experience and allowed us to be better skilled in conducting a modelling project from A to Z. We however failed to implement a few elements before the submission

of our predictions which prevented us from obtaining a very good score, such as a better treatment of missing values for the target variables (see Fig3.). Indeed, the linear interpolation gave us a big number of values which surely did not reflect the true ones, and we should have thought of another way, for example clipping the train set for this variable and taking only the observations after the big gap.

After the deadline, we were able to improve our results a bit, by taking for example the log of each target variable in order to diminish the effect of outliers. We realized the importance of treating well the initial variables and that adding a lot of data that does not necessarily translate into better forecasting. We were happy with the structure of our work and of the notebooks, as the classes and functions created were reusable easily and allowed us to try many different things.

Conclusion

Overall, this project was very taughtful as we learnt the importance of treating data before modelling and understood better the various techniques usable for treating time series. The deep learning models turned out to be a bit disappointing, even though some more time could have allowed better research on the best way to implement them. The main takeaway will be to explore better the variables before jumping to adding external data, and selecting the latter better as to avoid potential correlations between variables.

Appendix

Fig1.

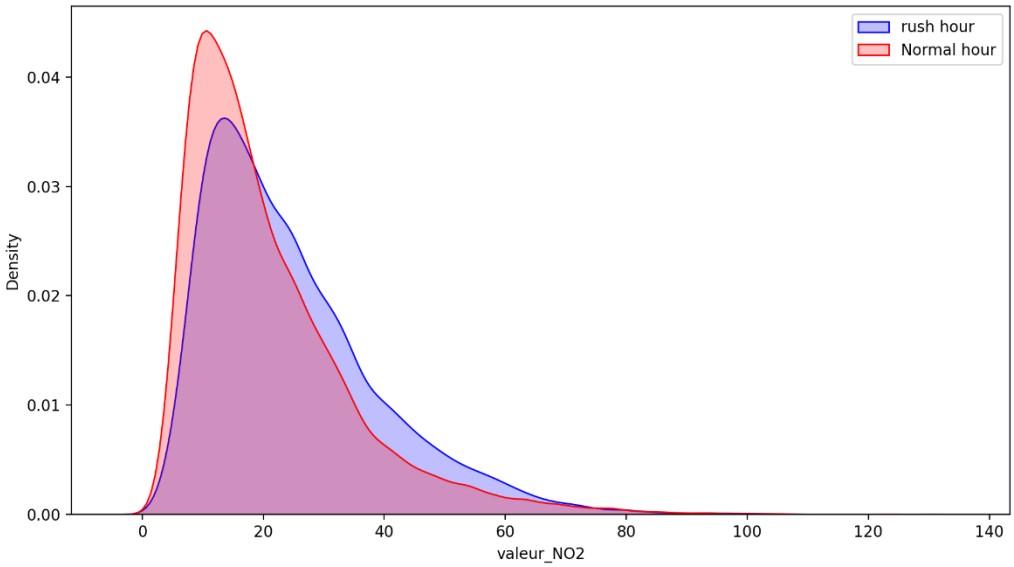
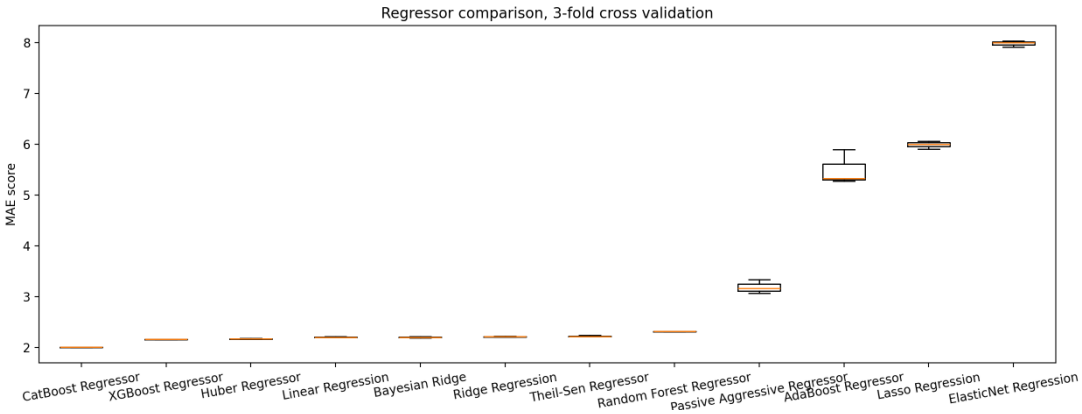


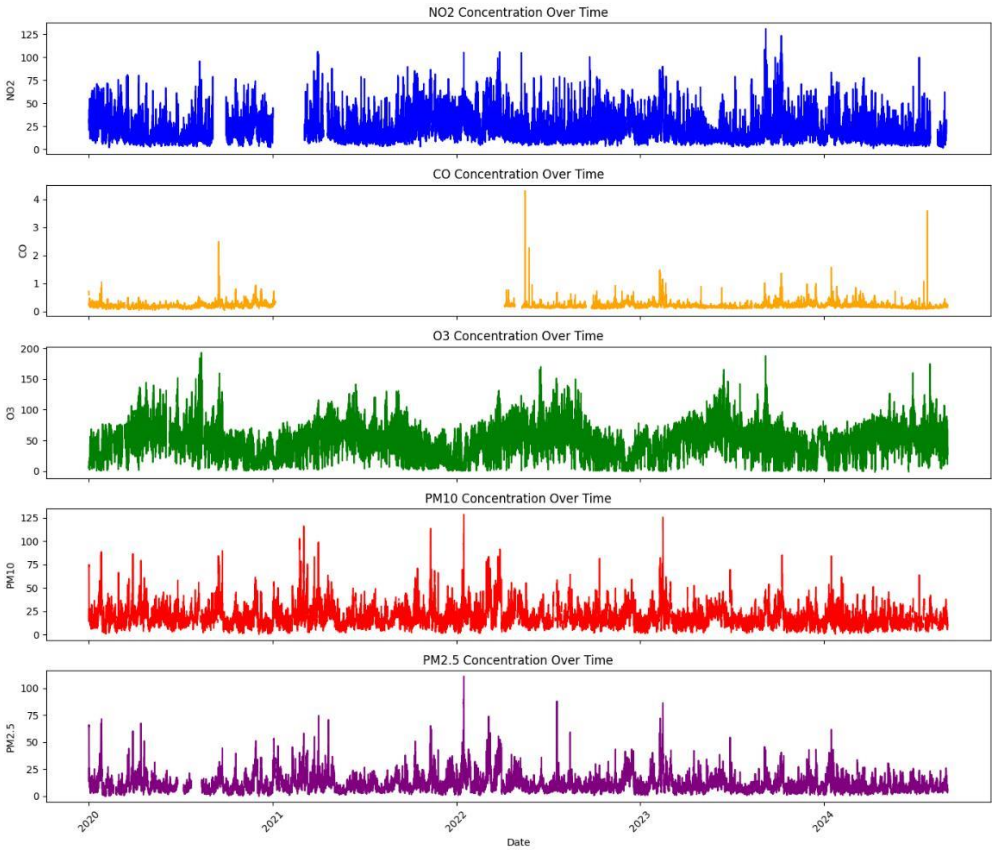
Fig2.



	Regressor	Sample size	CV folds	Mean MAE score	MAE score SD	Run time (min.)
7	CatBoost Regressor	10000	3	2.002	0.0028	1.91
6	XGBoost Regressor	10000	3	2.158	0.007	0.25
9	Huber Regressor	10000	3	2.169	0.0083	2.82
0	Linear Regression	10000	3	2.202	0.0088	0.0
8	Bayesian Ridge	10000	3	2.202	0.0088	0.01
1	Ridge Regression	10000	3	2.213	0.0069	0.0
11	Theil-Sen Regressor	10000	3	2.222	0.0083	6.01
4	Random Forest Regressor	10000	3	2.314	0.005	11.18
10	Passive Aggressive Regressor	10000	3	3.47	0.4544	0.03
5	AdaBoost Regressor	10000	3	5.638	0.1919	3.62
2	Lasso Regression	10000	3	5.991	0.066	0.01
3	ElasticNet Regression	10000	3	7.985	0.0521	0.01

Fig3.

Target variables before any treatment



Target variables after filling missing values with linear interpolation