Estimation of missing meteorological data

Project report by Popov Nicolai

1 Introduction

The main goal of the project is to estimate meteorological parameters in a place using historical measurements and synchronous data from several other places. In the provided dataset there are data for 5 meteostations that monitor weather in the following cities: Paris, Brest, London, Marseille and Berlin.

We designed and studied two different scenarios where the data from one station is missing:

1. At some point station in Paris breaks down and cannot produce measurements in real-time during a time period after which it is fixed and continues working. So real-time weather measurements will be absent in this first scenario. As far as numerous industry sectors rely on real-time weather measurements as well as weather forecast, a solution for inferring unrecorded or lost weather measurements would be profitable and relevant. More precisely, we consider the station's malfunction time to be 24h and we need to provide hourly temperature estimates during this period of time. We use previously recorded data from Paris station (before the breakdown) and real-time data from other stations for the modeling.

2. Another possible scenario is that data recorded in the past was lost from storage, so it obviously cannot be remeasured. In this scenario we consider the missing period to be 1 year instead of 1 day. It makes the problem more complicated since it is no longer possible to rely much on historical data for the Paris station. The temperature during the year varies a lot and using data from other stations is necessary to have good estimates.

To be able to measure the quality of solution we emulate the breakdown by removing the required period from the dataset and then comparing the estimates with recorded values of temperature.

Temperature is chosen as variable to predict because it is one of the basic and necessary information that people need on daily basis. This is not a constraint of the further proposed approach, because it can applied in a straight-forward manner to each weather variable measured by the weather station.

2 Dataset description

Weather variables presented in the datasets for each of 5 cities have the following names: skt, u10, v10, t2m, d2m, tcc, sp, tp, ssrd and blh. The time step is one hour and the length of records is 40 years between year 1980 and year 2019 inclusive, except for Paris dataset that includes year 2020 too. According to the website [1], the meaning of the weather variables is the following:

- blh - Boundary layer height
- d2m - 2m dewpoint temperature
- skt - Skin temperature
- sp - Surface pressure
- ssrd - Surface solar radiation downwards
- t2m - 2m temperature
- tcc - Total cloud cover
- tp - Total precipitation
- u10 - 10m u-component of wind
- v10 - 10m v-component of wind

One of the possible sources of uncertainty in the regression problem we have is noisy information that appears due to the noise in the measurements performed by sensors. On the other hand, missing information is not a problem because there are no gaps in the data provided which are also considered reliable. All the five datasets will be used for model preparation. Selected features and target values will be described below for each model. The last year 2019, which is present in the datasets for all cities, will be used as the test set. The 39 years before it, i.e. the ones in the range 1980-2018, will be used for training of the models and their evaluation on cross-validation.

Let's look on visualizations of some weather variables in Paris. They are plotted below during time periods of different lengths equal to one year, one month and one day.

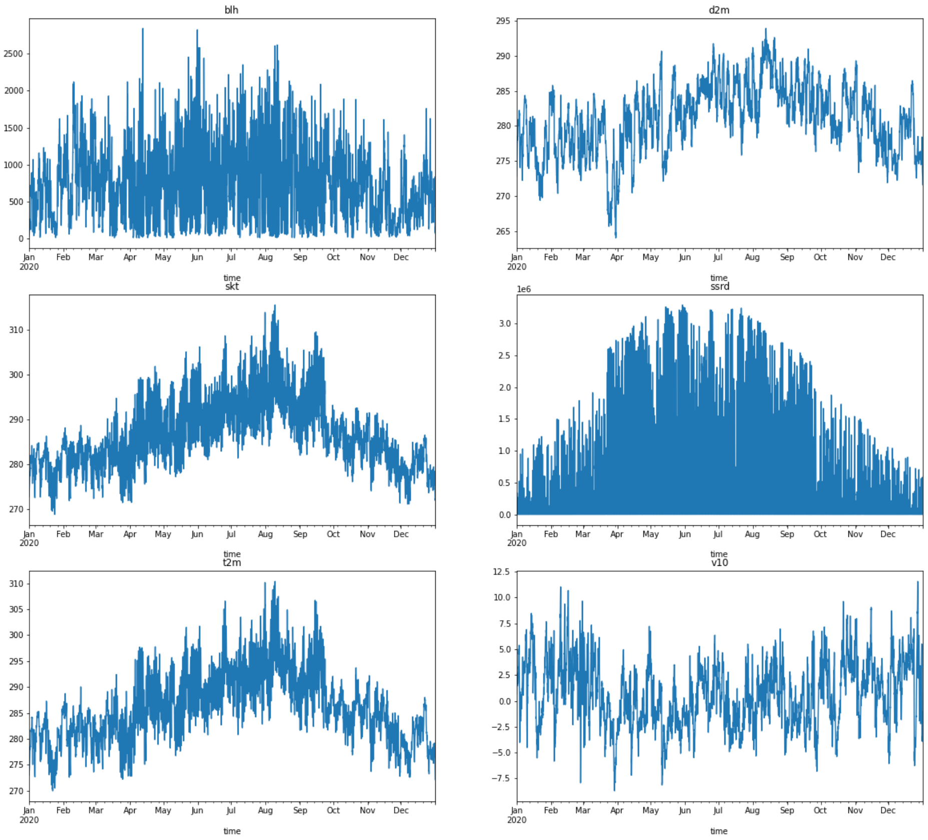

Figure 1: blh, d2m, skt, ssrd, t2m, v10 variables during 1 year long period in Paris

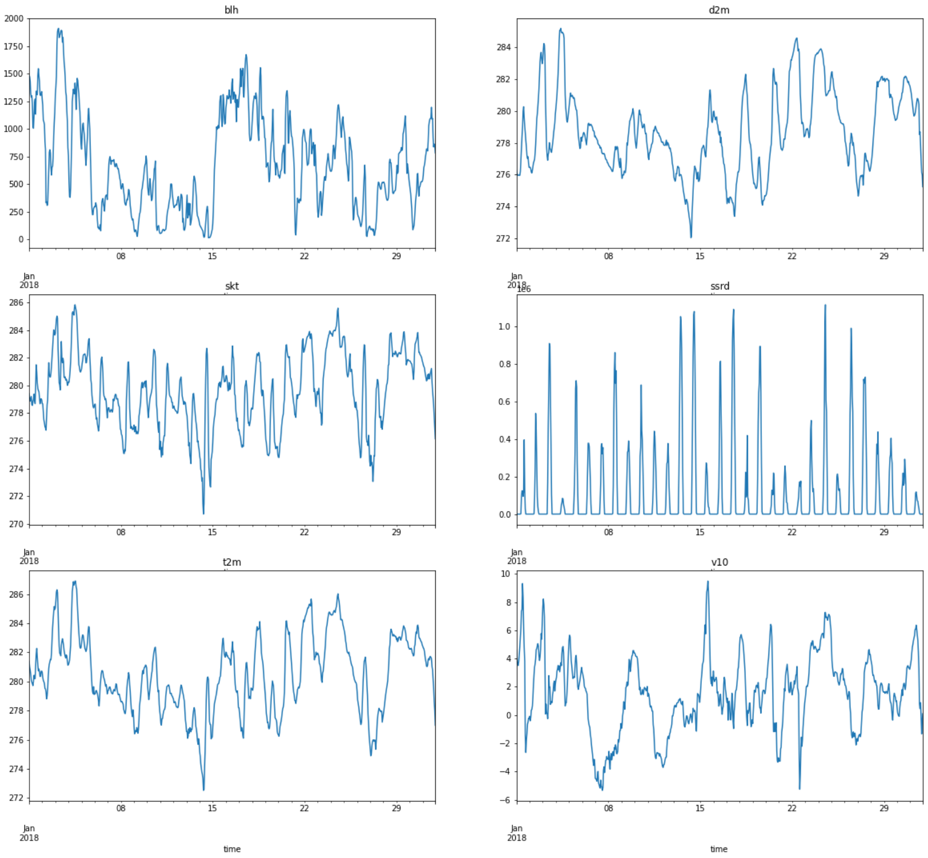

Figure 2: blh, d2m, skt, ssrd, t2m, v10 variables during 1 month long period in Paris

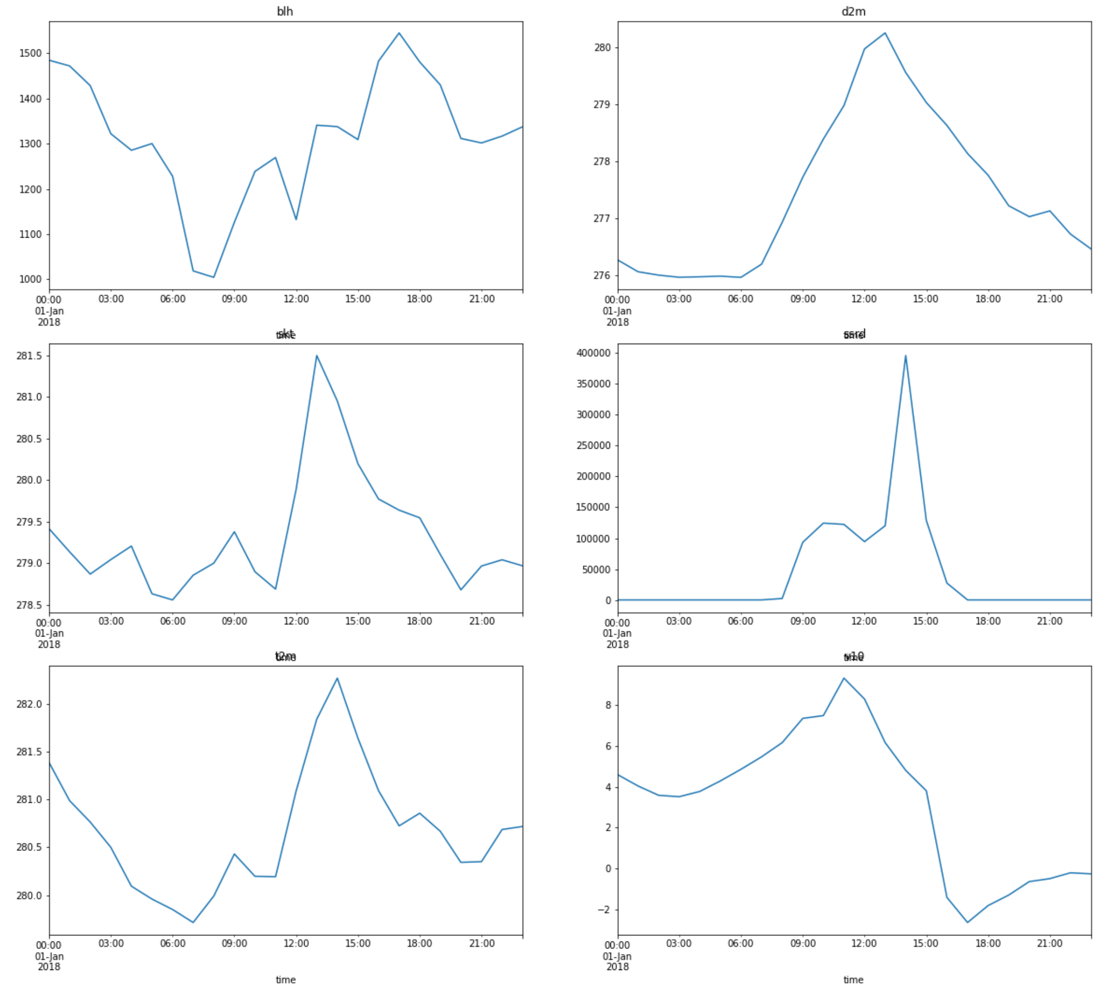

Figure 3: blh, d2m, skt, ssrd, t2m, v10 variables during 1 day long period in Paris

«t2m» weather variable has periods equal to one year and 24 hours. The same fact is true for several other variables. The properties of the other weather variables will be described in sections below where they are relevant.

3 Used methods

Machine Learning methods can be used to solve the defined above problem which is actually a regression problem. The real-time data from 4 weather stations outside Paris, that work properly, and the past data from station in Paris, which was recorded before it stopped working, will be used as features for the models.

The two types of regression models were used: Linear Regression with L1 regularization (Lasso) and regression algorithm called gradient boosting on decision trees from XGBoost library. The first one is chosen as baseline because it requires little time to fit and has only one hyperparameter to tune, namely regularization weight. The second one is considered to be among the best classic regression methods.

To realistically assess the model performance we use cross-validation. In each split of dataset the validation set follows the train set which is important because the data we work with is ordered in time. For each split the model is trained on the train set and then a mean absolute error (MAE) is calculated on the validation set. The number of splits is 10, so in total we obtain 10 values of MAE for training and evaluating the model on different train and validation data. Then we can assess the model using average of the 10 MAE errors and their standard deviation. MAE is used instead of another default error called root mean square error (RMSE) because it is more intuitive and is less influenced by the rare outliers in the prediction.

The total number of years in dataset is 39 years, so in cross-validation splits we make validation folds to be 1 year long and training folds - 29 years long. Validation fold follows the train fold in each split and their lengths in each split are equal to 254040 and 8760 hours respectively. The last 40th year is not used in any way except for the evaluation of the model on the test set. The same cross-validation and test set are used for evaluation of all models in this work.

In the text below the methods will be described for each of the two presented problems.

3.1 Problem №1

3.1.1 Linear Regression

Linear regression version used in this project is Lasso, so it has L1 regularization term in the loss. The L1 norm of weight vector is added to the mean square error of predictions. This regularization helps selecting features, because such a loss tends to zero some weights of the features that are not important.

Feature selection is performed by calculating the correlation between target variable «t2m» and all other weather variables with different lags, i.e. their shifted values. Only the ones with high enough positive correlation (greater than 0.2) are selected. This prevents using redundant features that in case of linear regression can decrease the regression performance. From the other side, this means that feature interactions were ignored during feature selection which is not the optimal strategy as impact of sets of features on target is not investigated.

Let's look on the correlations between target variable and all lagged variables per city:

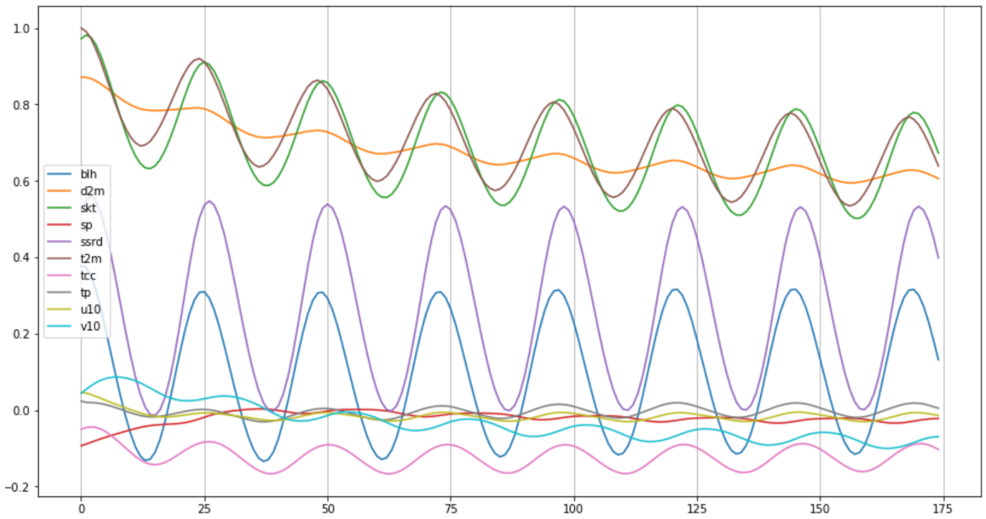

Figure 4: Correlations between «t2m» in Paris and all lagged variables in Paris for different lags.

In Paris dataset we see high positive values of cross-correlation between variables 'blh',

’d2m’, ’skt’, ’ssrd’ and ’t2m’ and auto-correlation of ’t2m’. Also the maximum correlations correspond to lags of 0, 24, 48 hours, etc.

Let’s do the same for another city - Berlin.

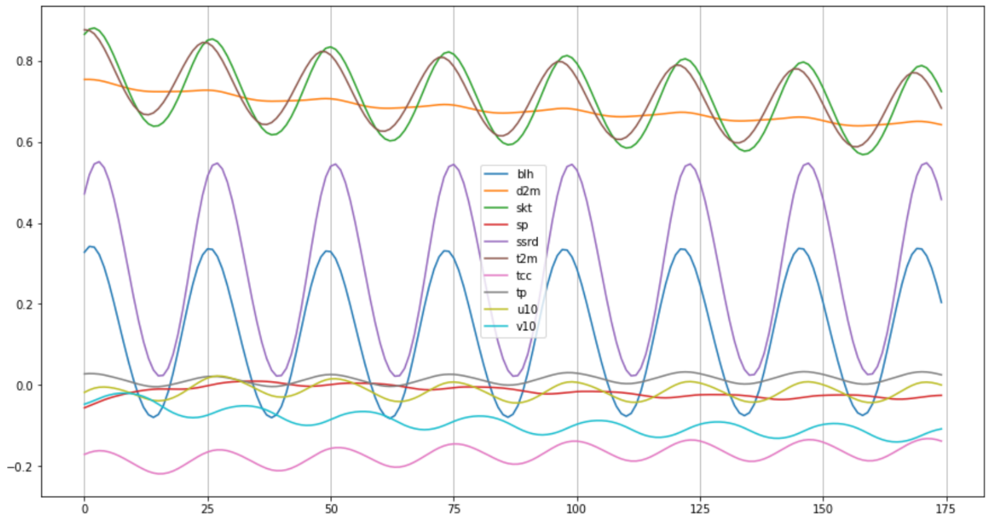

Figure 5: Cross-correlation between «t2m» in Paris and all variables in Berlin

Again the same 5 variables in Berlin (’blh’, ’d2m’, ’skt’, ’ssrd’, ’t2m’) have high positive correlation with ’t2m’ in Paris.

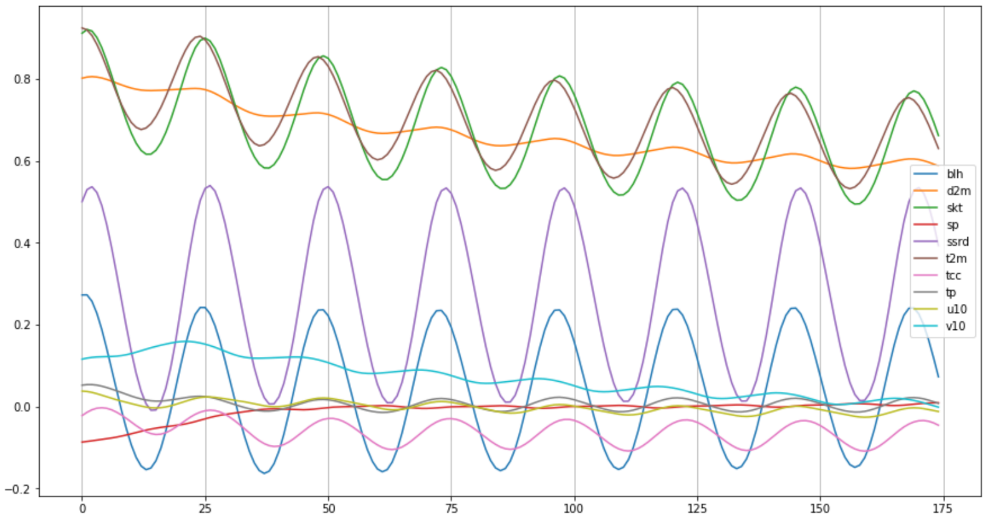

Figure 6: Correlations between «t2m» in Paris and all lagged variables in London for different lags.

The same 5 variables in London (’blh’, ’d2m’, ’skt’, ’ssrd’, ’t2m’) have high positive correlation with ’t2m’ in Paris.

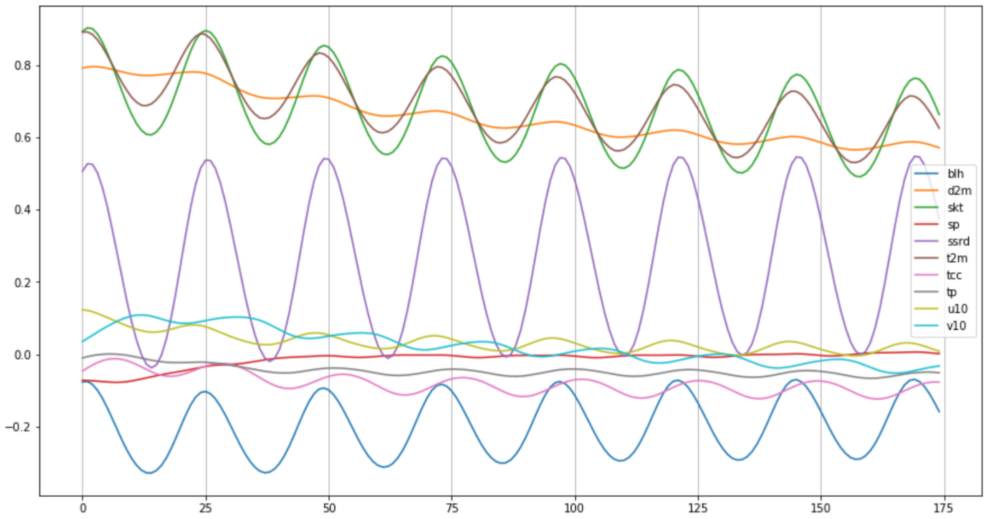

Figure 7: Correlations between «t2m» in Paris and all lagged variables in Brest for different lags.

For Brest only the 4 variables are selected ('d2m', 'skt', 'ssrd', 't2m'), because 'blh' correlation becomes negative.

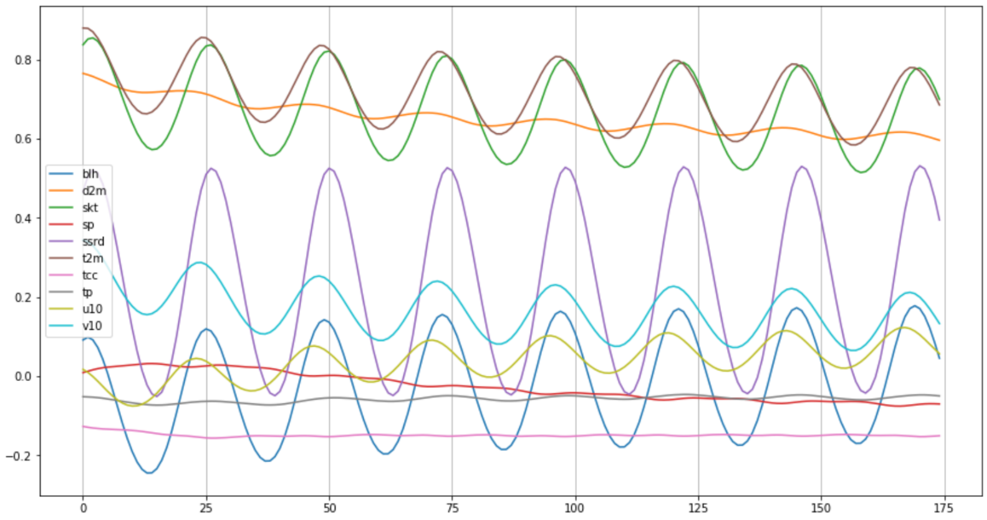

Figure 8: Correlations between «t2m» in Paris and all lagged variables in Marseille for different lags.

For Marseille only the 7 variables are selected ('blh', 'd2m', 'skt', 'ssrd', 't2m', 'v10', 'u10'), because wind components 'v10' and 'u10' start to correlate with 't2m' in Paris.

The variables listed above for each city are the ones that are selected from their datasets to be used in feature vectors to infer target variable. Their values will be taken at 8 different time moments, namely in the exact time of prediction, i.e. real-time values, and with a step of 24 hours back during the 7 days before the prediction moment. These features taken during the week at the same hour as the hour of prediction positively correlate with target feature and will hopefully add more information about the evolution of variables over time.

Scaling is necessary to perform on features for linear regression because they all are combined in a dot product with weight vector, so they should have a similar scale to be able to give equal contribution. Thus, MinMax scaling was applied to all features to cast them into [0, 1] interval. The min and max values calculated on train set, i.e. the first 39 years, are saved to use later for scaling of the test data, i.e. the last year.

Parameter tuning is the next step performed. Here the optimal weight alpha for the regularization term in the loss, i.e. L1 norm of weight vector, is found. For this purpose the grid search with cross-validation (CV) is used. As result, for each parameter value we have average value and standard deviation of MAE calculated over 10 splits of CV. So we can choose the best parameter, based on robust estimate of the error calculated using CV with train and validation folds similar to ones we have at evaluation on test set. The best parameter found for Lasso regression on CV yields MAE equal to 1.493 +/- 0.065 in degrees. Low variance of MAE shows that for each split in CV error does not differ too much. This result seems to be good for usage in practice. Next, the model with optimal parameter is fitted on the whole train set and evaluated on the test set: MAE is equal to 1.44 degrees (Kelvin or Celsius) at test set.

3.1.2 Gradient boosted decision trees

All weather variables are used from the 5 datasets for XGB regressor, because important **features are selected** in decision tree by its design. The lags are equal to 0, 24, 48 hours etc. for the 8 days starting from the hour of prediction, but for Paris the «t2m» values with lag 0 is what we predict, i.e. target. Another unnecessary step is **feature scaling** because the splitting in the nodes of a tree is performed by comparison and scale is not important for this operation.

Firstly, xgboost regressor with default parameters was evaluated on cross-validation and obtained MAE equals 1.459 +/- 0.072. This is a little bit better than the error achieved by Lasso regressor after feature selection and parameter grid-search. This proves that gradient boosted trees algorithm is far more powerfull than linear regression.

Parameter tuning is performed to improve performance of the model. Grid search with cross-validation is used, but this time there are a lot more parameters. Parameters that were tuned are considered to be the most important ones and were the following:

- max_depth : Maximum tree depth for base learners,
- n_estimators : int Number of trees to fit,
- gamma : Minimum loss reduction required to make a further partition on a leaf node of the tree
- min_child_weight : Minimum sum of instance weight needed in a child

Firstly, n_estimators is tuned alone, than max_depth and min_child_weight are tuned together because they control the trees and, finally, the gamma parameter. All other parameters can be tuned, but this tuning requires a lot of GPU computations, especially for high n_estimators values. MAE for the best parameters found is equal to 1.26 +/- 0.05 on cross-validation. Then, model with the best parameters is fitted on the whole train set and yields MAE equal to 1.17 in degrees on test set.

3.1.3 Results

Mean Absolute Error is a score that helps us evaluate the performance of the models quantitatively but we can also qualitatively assess the accuracy of our models by comparing original values of target variable, which is «t2m» in Paris, with predicted ones for the test set:

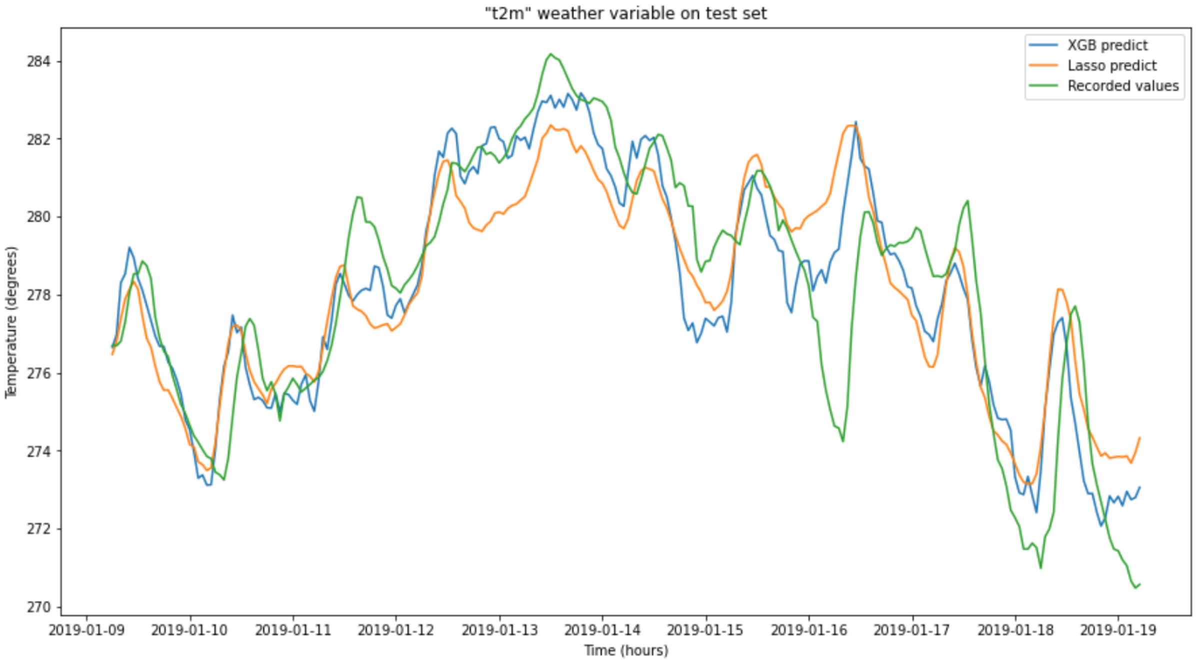

Figure 9: Original and predicted values of «t2m» variable in Paris on test set (Problem 1)

The Mean Absolute Error in degrees for predicting «t2m» temperature in Paris is presented in the table below for Problem 1:

Model / Data	Cross Validation	Test set
Lasso	1.49	1.45
XGB	1.26	1.17

We can notice than test error is lower than error on cross-validation. Probably this happens because the train set size for test evaluation is 39 year, but in cross validation train split includes 29 years. The best model allows us to infer «t2m» values that differ from ground truth at about 1.2 degrees (Celsius or Kelvin) and that is pretty decent for many applications.

3.2 Problem №2.

For this problem only XGB regressor was trained as it performs better than linear regression models. Also this model was used to assess the impact of different features on regression accuracy.

3.2.1 Feature engineering

As far as we don't have recent Paris data in our feature set in Problem 2, we can invent some new features to try to boost performance of the same model.

As we mentioned before, «t2m» variable has periods equal to one year and 24 hours. This means that «t2m» values are highly dependent on the time during the day and the month during the year. Hour and month values have gaps between 23 and 0 hours and numbers months - 12 and 1. But this time values are cyclic, because the first and last months in the year cycle are near each other and first hour follows the last one in day cycle. We can make this values cyclic, for example, by applying cosine functions to their values, but we prefer turning them into splines vectors to make them cyclic as proposed in web-source [2].

Another feature that probably will be useful is the Mean value of t2m temperature that is averaged over month before the prediction date in the previous year, because it could help predict the weather during day under examination this year. The sliding window size was chosen to be equal to 28 days after looking on the plots for several consecutive years below:

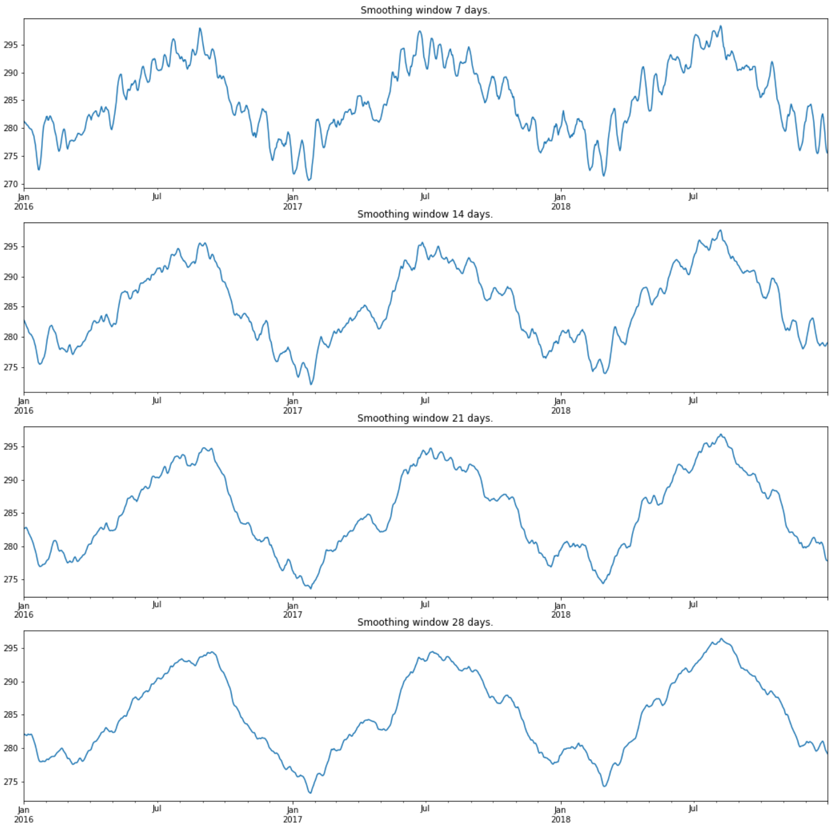

Figure 10: «t2m» variable in Paris smoothed with different windows

We see that 't2m' value still varies from year to year even after smoothing with 28 day long window. Mean monthly values of 't2m' will be calculated over one previous year for each day in the current year with 28-day-long window to be used as feature.

Parameters of Auto Regression Integration Moving Average model ARIMA fitted for 't2m' variable over a certain period of time will describe this weather variable as time-series evolving through time. Parameter of ARIMA (p,d,q) are the order of the model for the auto-regressive, differences, and moving average components. These 3 parameters for ARIMA model are chosen before fitting it on 't2m' weather variable. The intuition for parameter selection is similar to the approach described in sources [3] and [4].

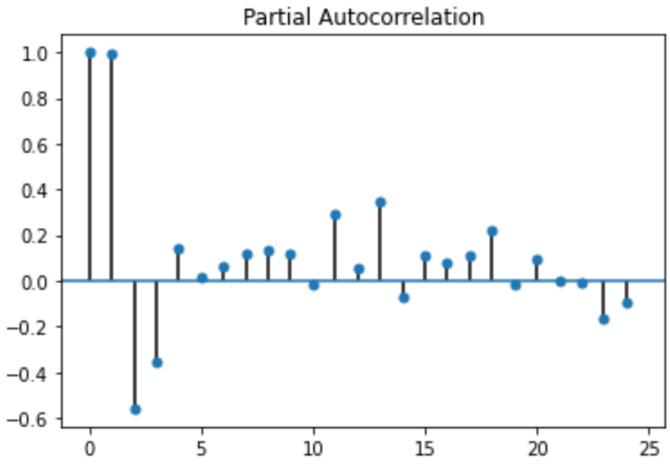

Figure 11: Partial Auto Correlation for «t2m» variable in Paris

Thus, the best order of the Auto Regression (AR) model is 3, because PACF values are high enough for lags 1, 2, 3 and then decreases abruptly.

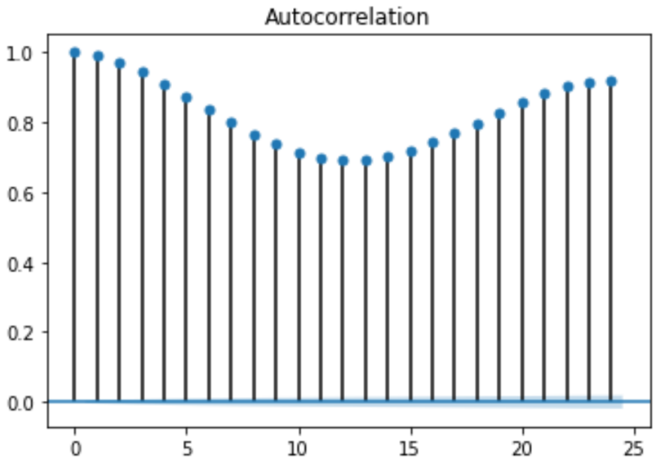

Figure 12: Auto-Correlation Function for «t2m» variable in Paris

Let's set the order of the Moving Average (MA) model is 10, because ACF values are high enough for the first 10 lags and value of 10 is not too big. Order of differences is set to value 2 because it is maximum possible value in the python library used. Of course, 24 is the best option for this, because it is the period for «t2m» weather variable. ARIMA with chosen parameters (3, 2, 10) fits the time series very well with MAE equal to 0.419:

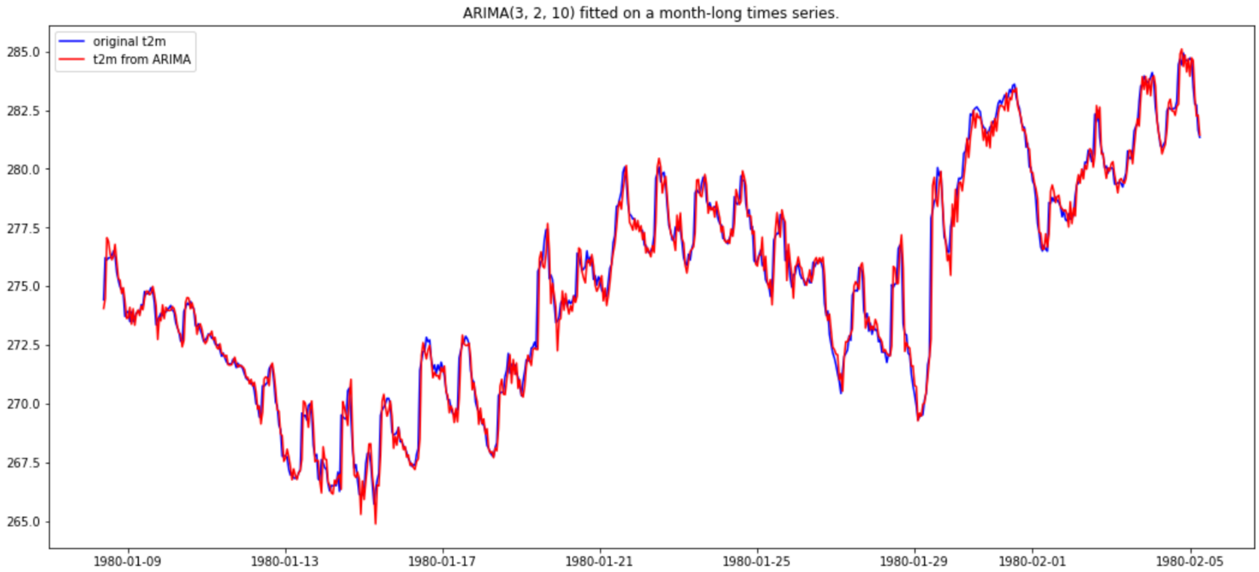

Figure 13: ARIMA fitted on 1 month long period of «t2m» variable in Paris.

As we can see ARIMA fits time series very well. Parameters from its Auto Regression part and Moving Average part are vectors of lengths 4 and 10 respectively. They could serve as features for prediction «t2m» in a current month after fitting the ARIMA on the same month one year ago, because these weights encode the pattern of behavior of the «t2m» time-series. One month interval comprises the change of t2m trend during the year, i.e. there are both decreasing and increasing slopes near local extrema in one month period.

3.2.2 Feature importance

Let's add some features extracted from past weather recorded in Paris to see if error can be reduced with the same XGB regressor parameters. Difference between Problems 1 and 2 is in their feature sets, because data recorded in Paris at time moments close to the moment of prediction, i.e. during the previous week, is not available in setting of Problem 2. Let's estimate the importance of these removed features by looking on the error of XGB regressor with default parameters trained with new reduced set of features. The same thing is done with the new added features that are described above in Feature engineering section.

To get a more representative comparison the cross-validation errors were assessed instead of test errors. For Problem 1 with recent Paris data among features we had MAE = 1.459 +/- 0.072 on CV for default XGB. And now for Problem 2 without recent Paris data among features we have MAE = 1.536 +/- 0.069 on CV for default XGB. The difference is less than 0.1 degree that means that the real-time information from other cities is enough to infer «t2m» temperature in Paris. On the other hand, we have MAE equal to 1.505 +/- 0.067 after adding new features. These means that these new features add some valuable information to the dataset but again we see that the major part of information necessary to regress «t2m» in Paris is contained in the real-time weather variables from other cities. MAE in degrees for «t2m» prediction in Paris using XGB regressor with default parameters is presented in the table below for the 3 mentioned feature sets:

	Cross Validation MAE
feature set 1	1.459 +/- 0.072
feature set 2	1.536 +/- 0.069
feature set 3	1.505 +/- 0.067

In the table above «feature set 1» are the features from Problem 1 that include recent Paris data, «feature set 2» are the features obtained after removing recorded Paris data from «feature set 1». The last «feature set 3» is the «feature set 2» with the following new features added: cyclic time, mean over the month a year ago and coefficients of ARIMA fitted on the month a year before prediction moment. The same XGB regressor model and the same folds in cross-validation were used to assess feature impact. Values of MAE shown in the table are in the format «*mean +/- standart deviation*» and both estimates are computed over splits of cross-validation.

3.2.3 Results

Among hyper-parameters of XGB regressor only n_estimators was tuned with cross-validation, but the same parameter tuning scheme can be applied here as it was in Problem 1. Also the optimal parameters found in Problem 1 were used in this problem and yielded the best score: MAE on CV equal to 1.324 +/- 0.056 in degrees. Finally, model was trained on the whole train set and evaluated on test set with MAE equal to 1.221. For qualitative assessment there is a visualization of the performance of the latter model below:

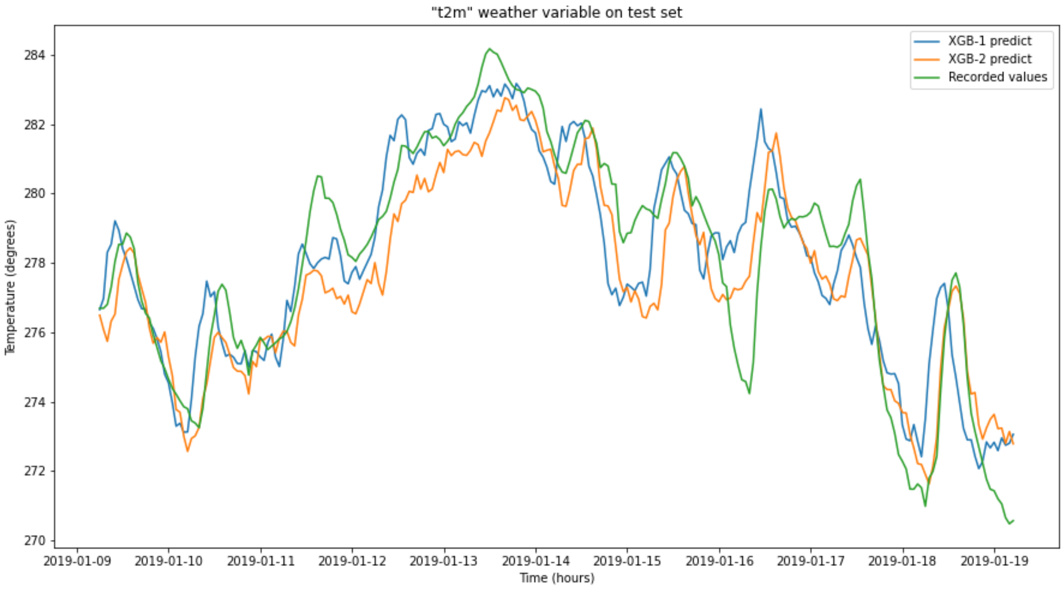

Figure 14: Original and predicted values of «t2m» variable in Paris on test set (Problem 2)

The Mean Absolute Error in degrees for predicting «t2m» temperature in Paris is shown in the table below for Problem 2 and compared with results from Problem 1 for the XGB model with the same parameters:

Model / Data	Cross Validation	Test set
XGB-1	1.26	1.17
XGB-2	1.32	1.22

We see that performance of XGB in Problem 1 («XGB-1») is slightly better than for «XGB-2» in the Problem 2 although they have the same parameters. This is the consequence of using different features to train these regressors. Again, the best model allows us to infer «t2m» values in Problem 2 with error equal to about 1.2 degrees (Celsius or Kelvin) which is quite adequate for use in practice.

4 Conclusion

In this project we worked with data from 5 weather stations in different European cities. The goal of this project was to create a model that could replace a station in Paris in case it stops measuring weather variables. The full ML methodology was used to train two classic ML algorithms for regression problem including feature selection, feature engineering, parameter tuning and proper evaluation. Obtained results look adequate for real life applications with weather measurements. In the future work, problem can be extended to a more complex one when all weather variables should be inferred on hourly basis and not only one. However, this problem can be solved simply by applying the proposed methods with one target variable for each variable separately or by using other methods for multivariate regression.

References

[1] https://sonra.io/data-marketplace/era5-variables/

[2] Sklearn tutorial

[3] https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7

[4] https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/