AML Challenge 1

ABESSERA Zachary
CHAPPAT Quentin
COLIN Laure-Amélie
MAILLARD Lucien

Table of content:

I. Exploring the dataset / Feature selection
II. Cleaning the data
III. Creativity
IV. Choice of models
V. Optimization of hyperparameters
VI. Conclusion

I. Exploring the dataset / Feature selection

Our work began by the exploration of the data : we had a deep look at the data to understand the correlation of the parameters with the temperature we had to predict. Here are some important conclusions revealed by the plots, such as the **correlation matrix plot** :

- The features named “**gfs_temperature_i”** where i is a number between 10000 and 97500 are highly correlated to the temperature we need to predict at 2 meters above the ground. Those parameters are the temperatures at different vertical levels and therefore are very important for our predictions, which is quite normal.

  During our studies, we have learned about the atmospheric model : $\boxed{T(z) = T_0 - a * z}$

And we decided to inspire ourselves to elaborate our first model : we started with a basic linear regression for which we kept only the “gfs_temperature” parameters. But it was of course not sufficient to make good predictions on the testing set and we had to study in more detail the other parameters that are important to predict the temperature, such as the dew or the snow.

II. Cleaning the data

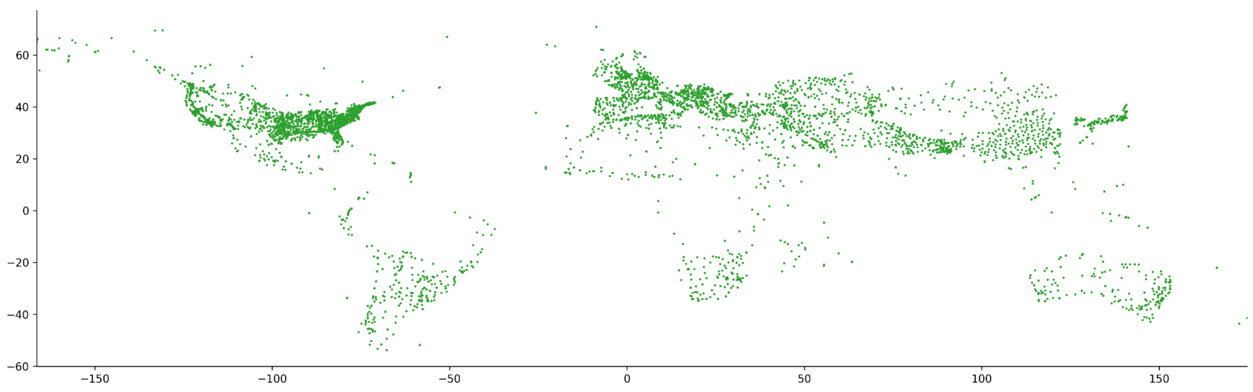
We had to perform some modifications on the dataset :

→ We **rename** the columns **longitude** and **latitude** because their names have been switched.

→ We **deleted the “out-domain test data”** (which are located in the train set in regions that are not at all present in the set to predict) because they are not useful since they will give data of uninteresting areas.

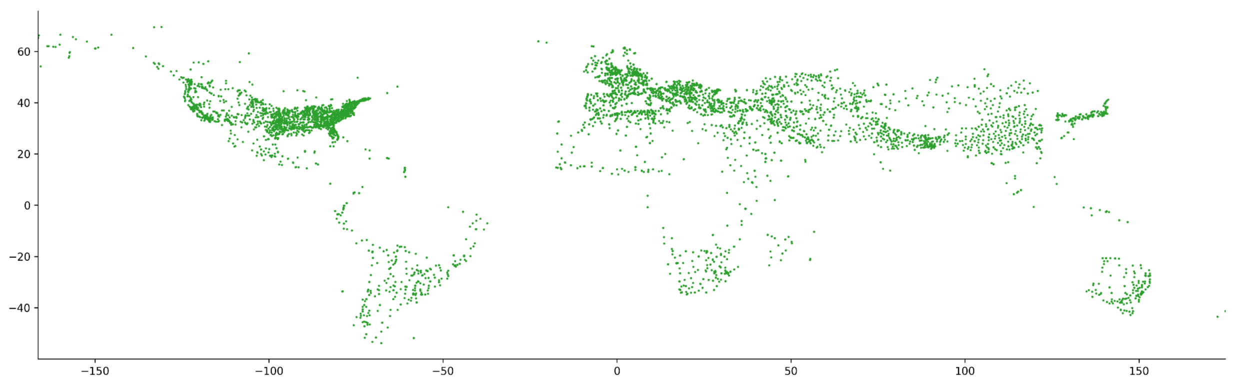
→ We found the rows that have “NaN” values and we decided to remove them as there are around 104,000 rows (which is around 5% of the test set) of this kind and because they are split evenly regarding the test set as this map shows:

Distribution of Nan in the test set worldwide



→ We found that around 361,000 rows are in double (same location and same time) but do not have the same information. We decided to delete them as well because we could not guess the real data and because, as for the NaNs, those data were evenly split around the world regarding the original test set.

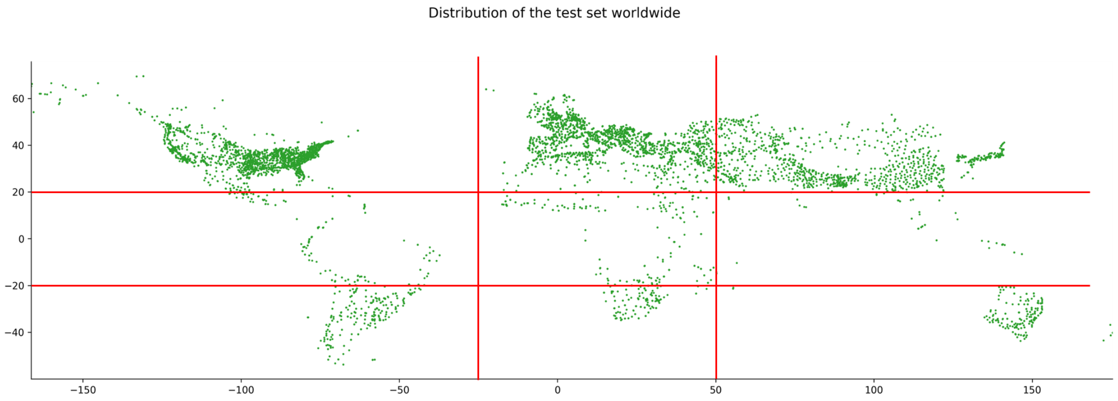
Distribution of duplicates in the test set worldwide



→ We decided to add monthly and daily mean temperature values (climate_temparature, wrf_t2_next, and cmc_0_0_6_2) for each element.
→ We converted the **date** information in classic format YYYY-MM-DD HH:MM:SS (and not UNIX) as the correlation matrix plot revealed to us that the date played an important part in the prediction of the temperature.
→ We **normalized** our data using similar codes to those given at the beginning of the challenge.

III. Creativity

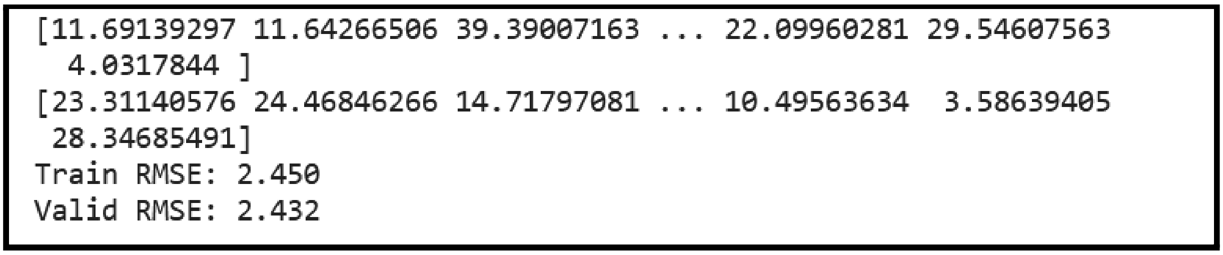
We realized that we have less data around the equator : therefore, we decided to **separate our dataset in 3 parts** (north, equator, south) and train our model one these. Afterwards, we even decided to separate the dataset in 9 parts to be more precise over the regions but we couldn’t achieve better accuracy so we kept the entire dataset.. Here is an example of the division of the training set into **9 datasets :**



IV. Choice of models

As we have seen during our courses, linear regression is particularly adapted for predictive analysis. It's very basic but it's easy and fast to train and it enabled us to have a strong control on our parameters.

For instance, for the dataset "north middle" (cf. section 1 for the division of the training set into 9 datasets), here are our results:



```
[11.69139297 11.64266506 39.39007163 ... 22.09960281 29.54607563
  4.0317844 ]
[23.31140576 24.46846266 14.71797081 ... 10.49563634  3.58639405
 28.34685491]
Train RMSE: 2.450
Valid RMSE: 2.432
```

At this moment, the parameters we considered are, of course, the **gfs_temperature_i** as we have seen their importance before but also other parameters (such as the sun_elevation and 'gfs_precipitable_water which represent the total precipitable water) we decided to add because of other works we have seen during our research and the correlation matrix plot.

We tried several ideas, like changing the number of parameters that were highly correlated to the temperature, or compute the correlation matrix for each 9 sub dataset.

1. Lasso

Lasso regression is a modification of linear regression. In Lasso, the loss function is modified to minimize the complexity of the model by limiting the sum of the absolute values of the model coefficients (also called the l1-norm). The Lasso model has an hyperparameter alpha to choose. With an optimized parameter, we could improve our results with a decrease of our public score on Kaggle leaderboard from 3,17 to 2,28.

2. XGBRegressor

Finally, we decided to try a more advanced model : **XGBoost**. Whereas linear regression is a parametric model, XGBoost is non-parametric and can approximate every function. Just as with linear regression, with the objective reg:squarederror, the goal of the algorithm is to minimize squared error. XGBoost is an implementation of the gradient boosted trees algorithm. Gradient boosting is a **supervised** learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

V. Optimization of hyperparameters

To find out the best hyperparameters for XGBoost, we apply the two most common algorithms: GridSearch and RandomSearch. We searched over every combination of specified parameter values. For instance, we specified possible values for **max_depth** and 3 for **n_estimators**, which are essential parameters in XGBoost.

Ex:

max_depth: [3,6],
n_estimators:[100, 200, 300]

VI. Conclusion

To conclude, this challenge showed us that **domain knowledge** is important for every machine learning project and that it is necessary to understand well manipulated data.

Therefore, our group decided to go for a **data-driven approach**, which had more sense for us than a model-driven approach. Finding out what is consistent, understanding different patterns in the dataset helps us to have a better understanding of how the data was generated and how to deal with those data such that a chosen model understands it.

Concerning the choice of models, we started by implementing a basic **linear regression and lasso regression** as we can have a great control of the given parameters and then, we decided to move towards to XGBoost, which has revealed to give the best results in terms of prediction.