Machine Learning for Earth and Environmental Sciences
Final project on moderate and extreme winds bursts in Europe

Fabien Augsburger[1]

Abstract

This project aims to predict a 3 hours prediction for 2 categories of wind in Europe (moderate and extreme convective bursts). Over the 4 algorithms used (Logistic Regression, Random Forest, Decision Tree, and Gradient Boosting), Random Forest performed the best overall.

1. Introduction

The goal of this project is to predict 3 hours in advance whether there will be an extreme wind burst or a moderate wind burst based on a dataset from 2011 to 2020 related to convective winds in Europe (fig. 1) with 10 predictors. It contains an index, the years, and 20 variables related to atmospheric data. Since there are 2 datasets, one for the moderate convective winds bursts and the other one for extreme convective winds bursts, the two datasets were merged and a binary column for the classification was created (1 for the extreme convective winds bursts and 0 for moderate convective winds bursts).

The link for the notebook is the following:

GitHub link

Here is a short description of the predictors:

- BSS_{0-1} and BSS_{0-6}: the bulk vertical wind shear between 0 - 1 km, and 0 - 6 km (expressed as a wind speed difference in m/s).
- $RH_{1000-850}$ and $RH_{700-500}$: the relative humidity, averaged between 1000 hPa - 850 hPa and 700 hPa - 500h Pa (expressed in %).
- $GUSTEX$: a variable used by Bart Geerts, which is proportional to the 500hPa wind speed (expressed in m/s).
- $LAPSE_{S1}$ and $LAPSE_{700-500}$: the lapse rate, defined as the difference in absolute temperature between 0 - 1 km divided by 1 km and between the 700 hPa and the 500 hPa surfaces and divided by the altitude difference between the 700 hPa and 500 hPa surfaces (expressed in K/km).
- $TOTAL_{TOTALS}$: The total totals index, defined as the temperature at 850 hPa plus the dewpoint at 850 hPa, minus twice the temperature at 500 hPa. This index is expressed in K and higher values correspond to stronger storms. More information on the total totals index.
- K_{index}: The K_{index}, defined as the following :

  $K = (T_{850} - T_{800}) + T_d850 - (T_{700} - T_d700)$

  Where T is the temperature at a xxx hPa and T_d is the dew point at xxx hPa.

  It is expressed in K and higher values correspond to a more unstable atmosphere, which increases the chance of a strong storm forming. More information on the k_{index}.

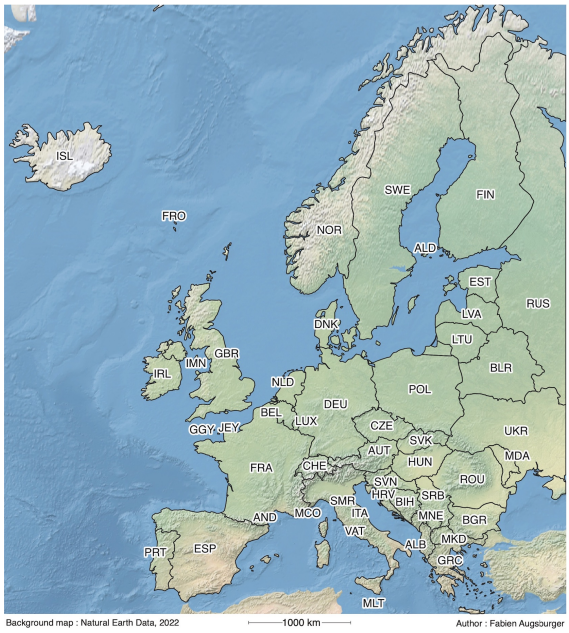

Figure 1. Zone of interest: Europe. Source: Natural Earth Data, 2022

- $CAPE$: The $CAPE$, or Convective Available Potential Energy, is defined as the indication of the instability (or stability) of the atmosphere and is expressed in J/kg. Higher values increase the likelihood of a strong storm forming. More information on the CAPE.

Description of two predictors for the moderate (top of table 1) and extreme convective winds bursts of the dataset (bottom of table 1):

Table 1. Description of the moderate (top) and extreme bottom (convective winds bursts dataset

DESCRIPTION	BSS0_1_MEAN	BSS0_6_MEAN
COUNT	20613.000000	20613.000000
MEAN	4.983855	12.815968
STD	3.274193	6.495572
MIN	0.027840	0.593803
25%	2.545131	7.865498
50%	4.379407	12.039471
75%	6.800914	16.911604
MAX	20.297779	43.586357
DESCRIPTION	BSS0_1_MEAN	BSS0_6_MEAN
COUNT	15337.000000	15337.000000
MEAN	6.818424	14.547084
STD	3.395588	6.289388
MIN	0.040366	0.949590
25%	4.404220	9.939423
50%	6.480618	13.968437
75%	8.945686	18.663570
MAX	21.888948	48.024220

2. Methodology and algorithms used

The method was inspired from Lagerquist, and al. (1). Thus, four algorithms are used:

- Logistic Regression
- Random Forest Classifier
- Decision Tree Classifier
- Gradient Boosting Classifier

First, a default run is made on each one of the classifiers, then a search for the best hyperparameters is used (via HalvingGridSearchCV) on Random Forest, Decision Tree and Gradient Boosting. The hyperparameters tuning is on

- `max leaf nodes`
- `min samples split`
- `max depth`

Some specific algorithms hyperparameters are also tuned: `n estimators` for Random Forest and `learning rate` for Gradient Boosting.

Then, for the performance of each algorithm, an accuracy test is done on the validation set by comparing the default settings and the custom settings. Afterwards, another accuracy test is done on the validation and testing set with the best model (based on its previous accuracy test).

The same process is repeated for the precision test. Finally, on each algorithm tuned, a permutation feature is tested to see which predictor is mainly used, and by which algorithm.

2.1. Logistic Regression

Parameters used for the first Logistic Regression are the default ones. Then a second test is conducted with the following `solver` : 'liblinear'. In the results, the 4 solvers that it supports roughly performed the same.

2.2. Random Forest

The same method used in Logistic Regression is applied for the Random Forest classifier. The second test is conducted with the following hyperparameters tuning (table 2).

Table 2. Hyperparameters tuning on the Random Forest Classifier

HYPERPARAMETERS	RANGE
MAX LEAF NODES	60, 80, 100, 120, 140, 160, 180, 200
MIN SAMPLES SPLIT	5, 10, 15, 20, 25
N ESTIMATORS	50, 100, 150, 200, 250
MAX DEPTH	3, 10, 20, 40

2.3. Decision Tree

Like the first two algorithms, a default run is done, then a hyperparameters tuning is done with the same one used in table 2 (except for `n estimators`).

2.4. Gradient Boosting

The hyperparameters tuning is done differently, as shown in table 3.

Table 3. Hyperparameters tuning on the Gradient Boosting

HYPERPARAMETERS	RANGE
MAX LEAF NODES	120, 140, 160, 180, 200
MIN SAMPLES SPLIT	5, 10, 15, 20, 25
LEARNING RATE	0.05, 0.1, 0.15, 0.2
MAX DEPTH	3, 10, 20, 40

3. Results

3.1. Accuracy

The results from the default and the custom model are the following (fig. 2). Those results are solely based on the validation set.

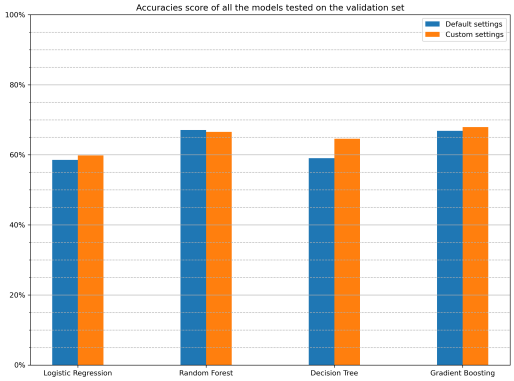

Figure 2. Accuracy of each algorithm on their default and custom settings

Since there are no models that clearly performed better in their default settings or custom settings, an accuracy test is then done with the testing set, and the validation set on the most accurate models. Results are displayed in table 4.

Table 4. Accuracy of each algorithm on the testing, and validation set

CLASSIFIER	SETTINGS	VALID	TEST
LOGISTIC REGRESSION	CUSTOM	59.84%	60.83%
RANDOM FOREST	DEFAULT	67.07%	67.65%
DECISION TREE	CUSTOM	64.57%	64.33%
GRADIENT BOOSTING	CUSTOM	67.92%	66.80%

3.2. Precision score

The precision score of each algorithm based on the validation set is the displayed in figure 3. It compares the default settings versus the tuned settings:

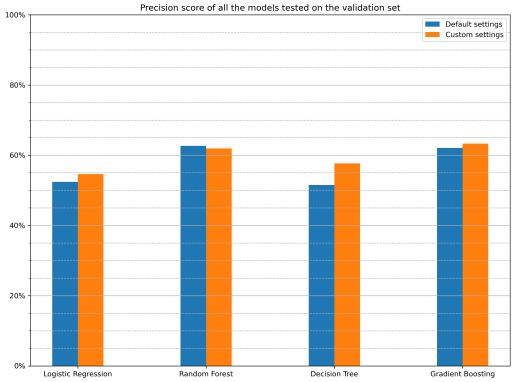

Figure 3. Precision of each algorithm on their default and custom settings

The precision table is the following (table 4):

Table 5. Precision score of each algorithm on the validation, and testing set

CLASSIFIER	SETTINGS	VALID	TEST
LOGISTIC REGRESSION	CUSTOM	54.62%	55.70%
RANDOM FOREST	DEFAULT	62.68%	63.21%
DECISION TREE	CUSTOM	57.68%	56.99%
GRADIENT BOOSTING	CUSTOM	63.30%	61.69%

3.3. Features permutation

The results for the permutation tests are displayed in figure 4.

4. Discussion

The results from the 4 classifiers with their modifications are not great. An accuracy of 57-66% can be improved with the use of a neural network (Lagerquist, and al.)(1). The precision score is not better and ranges from 49% to 63%. The best algorithm overall is Random Forest, mainly because the results with the validation and testing set are very close. Gradient Boosting comes second, because it has a slight overfit on each testing set. Third and last are respectively Decision Tree and Logistic Regression: Decision Tree performs better in the accuracy and precision tests and has fewer differences between the validation and testing set compared to Logistic Regression.

It is interesting to see that Logistic Regression use mainly one predictor, which is the mean CAPE (to a lesser extent, it also uses $RH_{1000-850}$ and $RH_{700-500}$). Gradient Boosting uses 3 others predictors : mean GUSTEX, mean BSS_{0-1}

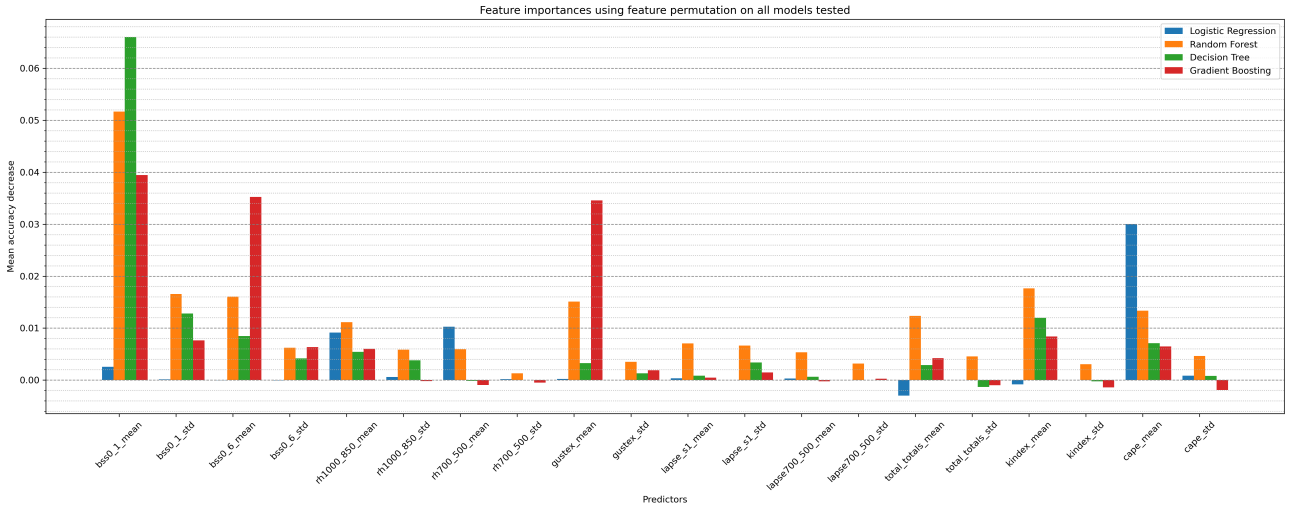

Figure 4. Feature importance of each algorithm based on the best version of the model

and mean BSS_{0-6}. Random Forest and Decision Tree both use mainly the mean BSS_{0-1}, but Random Forest is the only algorithm that uses most of the other predictors. Decision Tree also uses the standard deviation of BSS_{0-1} and mean K-index for its predictions. Knowing that mainly the CAPE and K-index are used (2)(3)(4) to predict those categories of wind, it is possible to use other variables, such as BSS_{0-1}, BSS_{0-6}, or GUSTEX with a relative accuracy.

Acknowledgements

I thank Iat Hin Tam for the collection of the dataset, and its pre-processed.

References

[1] R. Lagerquist, A. McGovern, T. Smith, Machine learning for real-time prediction of damaging straight-line convective wind 32 (6) 2175–2193. doi:10.1175/WAF-D-17-0038.1.
URL https://journals.ametsoc.org/doi/10.1175/WAF-D-17-0038.1

[2] G. P. Pacey, D. M. Schultz, L. Garcia-Carreras, Severe convective windstorms in europe: Climatology, preconvective environments, and convective mode 36.

[3] M. Taszarek, J. T. Allen, P. Groenemeijer, R. Edwards, H. E. Brooks, V. Chmielewski, S.-E. Enno, Severe convective storms across europe and the united states. part i: Climatology of lightning, large hail, severe wind, and tornadoes 33.

[4] M. Taszarek, J. T. Allen, K. A. Hoogewind, H. E. Brooks, Severe convective storms across europe and the united states. part II: ERA5 environments associated with lightning, large hail, severe wind, and tornadoes 33.