Report CNR Wind Power forecasting

L. Chenin

22/06/2020

1. Overview

This is a report on the challenge made by Compagnie Nationale du Rhone (CNR) - the French leading producer of exclusively renewable energy (water, wind, sun) - on hourly wind energy production forecast. This challenge was selected as part of the edX HarvardX: PH125.9x Data Science: Capstone Project Submission.

CNR currently owns around 50 Wind Farms (WF) for a total installed capacity of more than 600 MW. Every day, CNR sells on the energy market its wind energy production for the day ahead. In order to sell the right amount of energy, as well as for legal requirements towards the French Transmission System Operator (TSO) in charge of the electric network stability, CNR needs to know beforehand how much energy the wind farms will produce the day ahead.

see https://challengedata.ens.fr/participants/challenges/34/

The goal of this challenge is to predict the energy production of six Wind Farms (WF) owned by CNR. Each WF production will be individually predicted, using meteorological forecasts as input. Predictions will focus on the day-ahead energy production (hourly production forecasts from day D+1 00h to day D+2 00h).

The data set was retrieved after login at the above link and uploaded at

https://github.com/lchenin/CNR_challenge_2020

The project assignment I made from this goal is to elaborate two prediction models of the six Wind Farm train sets based on the arima algorithm from the R forecast package and the random forest algorithm from the R random forest package on the six Wind Farm test sets and compare the metrics they arrived at (77 and 80 respectively) to the benchmark metric of CNR (31).

As these figures are pretty high, a number of refinements have to be further studied to improve the model(s).

The report is structured as follows:

- Section 1 outlines the problem, describes the dataset and the key steps of the analysis;
- Section 2 checks the data and explains the prediction model used;
- Section 3 presents the modeling result and discusses the model performance;
- Section 4 concludes with a brief summary of the report, its limitations and future work.

Note: as this document is a report, no script or plot is included. The markdown document includes the report material, plots and the script split into chunks while the script is the R piece of code.

1.1. Problem definition & objective

The project assignment consists in forecasting wind production for the test period that runs from January the 16th of 2019 to September the 30rd of 2019 (8 months and 15 days), provided that data for the train period that runs from May the 1st of 2018 to January the 15th of 2019 (8 months and 15 days) is used. Evaluation will be performed by the challenge provider, CNR, on raw observed hourly WF power production. As a consequence, comparison will be made between this metric and CNR benchmark.

The objective I will follow is to run a prediction model and compare it to the metric value obtained from their submission to CNR.

The metric used to rank the predicting performance is a relative form of the absolute error. CNR, the challenge provider call it the CAPE (Cumulated Absolute Percentage Error). The formulation of CAPE for one WF would be the following:

$$\mathrm{CAPE}_k\left(\widehat{Y}_k, Y_k\right) = 100 \times \frac{\sum_{i=1}^{N_k}\left|Y_{i,k} - \widehat{Y}_{i,k}\right|}{\sum_{i=1}^{N_k} Y_{i,k}}$$

‘‘‘

with $CAPE_k$ is the metric for the WF k, N_k is the length of the test set for WF k,
$Y_{i,k}$ is the observed production for WF k and hour i (MW or MW.h) and

$$\widehat{Y}_{i,k}$$

is the predicted production for WF k and hour i (MW or MW.h).

For convenience reasons, data relative to the 6 WF have been regrouped in the same train and test input files. Therefore, the metric used in the challenge is the overall average CAPE for the 6 WF, calculated as:

$$\mathrm{CAPE}\left(\widehat{Y}, Y\right) = 100 \times \frac{\sum_{i=1}^{M}\left|Y_i - \widehat{Y}_i\right|}{\sum_{i=1}^{M} Y_i}$$

This formulation results in a non-homogeneous contribution of all the WF to the final value of CAPE: CAPE will be more sensitive to WF with the highest energy production values.

1.2. Dataset

The dataset provided is the combined six WF train and test data. The test production file is random and given by the challenge provider as a template for file submission.

the description made by the challenge provider is as follows:

ID: This is the unique ID of each row in the csv files. One ID correspond to a couple Time / WF. The ID of the test set are consecutive to the ID of the training set.

WF: The considered Wind Farm. WF ranges from WF1 to WF6. It is crucial for the competitors to be aware that this prediction problem is totally dependent to the WF considered. In other words, the statistical link between input variables and wind power production is completely different from one WF to another. Consequently, it could be judicious to train specific prediction algorithms for each WF, instead of training a unique algorithm which could be unable to model the behavior of each WF.

Time (UTC): date and hour of the target timestep, i.e. corresponding to the observed Power production. Time zone is Coordinated Universal Time (UTC).

Meteorological variables: Numerical Weather Predictions are provided by meteorological centers several times a day (updates), typically at 00h UTC, 06h UTC, 12h UTC and 18h UTC. We call these sets of

forecasts "Runs". Consequently, if the input file contains forecasts arising from several runs, this implies that a single NWP is associated with several forecasts for the same forecasting time. Therefore, the information on the hour of run is provided.

The format of the header of the csv files for the meteorological variables is the following:

NWPi_HourOfTheRun_DayOfTheRun_Variable

With NWPi the considered Numerical Weather Prediction model (meteorological model);

HourOfTheRun the hour (UTC) of the considered run. According to the NWP, it could be 00h, 06h, 12h and 18h (case of NWP with 4 runs per day) or only 00h and 12h (case of NWP with 2 runs per day);

DayOfTheRun the day of the considered run. We provide in the csv files predictions from the D_2 day runs (the day before yesterday), D_1 day runs (yesterday) and D day runs;

Variables of the different meteorological variables forecasted by the NWP. These are essentially U, V and T:

U and V components of the wind at 100m (or 10m) height (m/s): these are the zonal and meridional velocities of the wind, respectively. Both are given at a height of 100m above ground for NWP1, NWP2 and NWP3. U and V are given at a height of 10m for NWP4. Even if these variables are given at hourly timestep, we draw competitors attention on the fact that the temporal representativity of the given values is for a 10-minutes window ranging from H-10 min to H.

Additional remark: since wind power production is principally driven by the wind speed impacting turbines, it could be useful for the competitors to derive wind speed (and wind direction) from U and V. This can be done using a simple trigonometric calculation of the magnitude and direction of a vector with U and V components. The choice is let to the competitors.

Temperature of air (°C), abbreviated T: this is the averaged temperature over the entire hour (from H-1 to H). Wind power production is sensitive to air temperature since it affects the air density. This variable is provided only for NWP1 and NWP3.

Total cloud cover (%), abbreviated CLCT: this is the total cloud cover of the sky, ranging from 0% (clear sky, no cloud) to 100% (fully clouded sky). The value is an instant value at hour H. This variable is provided only for NWP4.

The data sets have been uploaded from https://challengedata.ens.fr/participants/challenges/34/ to https://github.com/lchenin/CNR_challenge_2020 in order to get them accessible when running an R script.

1.3. Key steps of analysis

The analysis is performed in the following sequence of steps:

- Data check for a review in terms of tidiness and completeness.
- Data split of the train and test sets into six WF train and test sets as prediction is recommended to be made on individual WF.
- Data exploration of these WF train and test sets, including visualisation of distributions and/or relationship between variables (or predictors) to serve as guidance for the development of an appropriate prediction model.
- Data aggregation of combined predictors, such as wind speed.
- Model development and running on the test set to derive figure of merits as per the challenge provider metric.

2. Analysis

2.1. Data check

The train and test data sets are conform to the description made by the challenge provider: X_train has 37,375 observations of 102 predictors labelled NWPi_xxyh_ with D_2 or D_1 or D followed bu _U, or _V, or _T or _CLCT as the case may be, and 3 variables, ID, numbered from 1 to 37,375, WF as a character "WFi" with i = 1 to 6 and Time, defined as a character, but identified as a date by hour ("01/05/2018 01:00" in the dmy manner) .

The same applies to X_test for teh same 102 predictors and 36,529 observations.

there is a lot of NA's that amounts to 46 % of the train set predictors, and 76 % of the test set predictors. As a consequence, 'beefing up' the test set with missing data is definetly an option to pursue.

We start by grouping the wind farms WF1 to WF6 and check the train set for their counts or numbers of observations. All have 6,239 counts excep WF5 for 6,180. for the test set, the same discrepancy appears, with all WF groupings at 6,190 except WF5 at 5,579.

By plotting the Y_test, we confirm it is effectively random and serves as a template for the prediction submission.

the hourly count of production shows a dominant portion for WF1.

2.2. Model development

As there is a lot of missing values in the 102 predictors, a first idea was to aggregate some of the predictors to get an estimation of the wind turbine production. As the wind speed to the power three is proportionnal to the wind production, we computed the 36 wind speed that can be derived from the 4 NWPs, and add them altogether to get a unique 'predictor'.

As this data project is clearly related to time series, we used Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. And particularly section 12.8 Forecasting on training and test sets, to fit an arima model to the above unique 'predictor'.

We split the data set, train and test into six wind farm specific train and test sets and apply an auto.arima algorithm from the R package forecast to the variable `tot`, that is the sum of the 36 individual production computed, as a first tool and then apply a random forest algorithm from the R package randomforest to the same variable `tot`.

All production is positive or zero, and maximum is limited in the wind farms by the turbine number and rating, obtained from the supplementary data set provided:

WF1 for 10.14 MW comprising 4 turbines of power 2.25 MW each, or 10.0 MW.

WF2 for 11.92 MW comprising 6 turbines of power 2.0 MW each, or 12.0 MW.

WF3 for 13.47 MW comprising 6 turbines of power 2.25 MW each, or 13.5 MW.

WF4 for 9.05 MW comprising 4 turbines of power 2.25 MW each, or 10.0 MW.

WF5 for 11.97 MW comprising 6 turbines of power 2.25 MW each, or 13.5 MW.

WF6 for 5.07 MW comprising 2 turbines of power 2.0 MW each, or 5.0 MW.

The model developed has not been constrained with these low and high values and this is a direction for improving the prediction.

3. Model development and performance

3.1. Arima

we start with wind farm WF1 for the first training set and make a plot of the variable 'tot' to look for similarity to the production train and apply the auto.arima algorithm and various order c(p,d,q) is to be applied to WF i = 1 to 6 to get a fitting model.

we observe that neagtive values are predicted! then we normalize X1_train_fit and compare/ correlate to Y1_train. Negative values will be set to zero.

Even if the peak is normalized, there are numbers of regions where the fit is not correct.

Now for the test part, we fit the arima model from the train set: and also normalize X1_predic for comparison / correlation to max(Y1_test) that is unknown => take max(Y1_train)

we repeat the above for the five wind farms WF2 to WF6.

We summarize the prediction in a single data frame and write the export file to sublit our validation to the challenge provider.

From the challenge provider metric, when submitting the above file, the score is 77, well above the benchmark of 31.7, the best score of the competitors being 29.6. A lot of improvements is to be looked for.

3.2. Random forest

the random forest is used on the 102+37 = 139 predictors

from the challenge provider metric, when submitting the above file, the score is 80, well above the benchmark of 31.7, the best score of the competitors being 29.6. However it seems a little less than Arima. Still a lot of improvements is required and is to be looked for.

4. Summary and conclusion

This project is an opportunity to discover the time series prediction domain, and practice R. I selected this challenge, as the data is not that heavy and workable, but I still have to find the tools to complement and assess the missing data in the train set (46 % of the predictors content is NA) so that in a first step, I can smartly complement the missing data in the test set (76 % of the predictors content is NA, and what if this percentage is decreased to 46 % as the challenge provider hinted?)

Another direction to go is to apply auto.arima to the Xi_train$Production to 'correct', so to speak the auto.arima then applied to the `tot` variable has been so to speak 'trained' on the train set.

For instance the ARIMA(5,1,5) from `tot` in WF1 corresponds ot an ARIMA(0,1,2) for `X1_train$Production`. Also the use of the sum of production, tot from the 36 hourly production might not be appropriate, and is not sanctionned by a dedicated regression.

For that matter, an idea to be explored could be to use the auto.arima to provide not so randomly generated figures to complement the predictor missing values.

On the other model used, the random forest does not present negative values but some how the minimum values are well above zero for WF2, WF3, WF4, which is not satisfactory. I am still left to further dig out this point.

And finally I have to browse through the numerous models of the caret package to explore other models to better my score, and combine models as combining results improve scores.