PREDICTING WIND ENERGY PRODUCTION WITH SCIKIT-LEARN

1. INTRODUCTION

Nowadays, electricity networks of advanced countries rely more and more in non-operable renewable energy sources, mainly wind and solar. However, in order to integrate energy sources in the electricity network, it is required that the amount of energy to be generated to be forecasted 24 hours in advance, so that energy plants connected to the electricity network can be planned and prepared to meet supply and demand during the next day (For more details, check "Electricity Market" at Wikipedia).

This is not an issue for traditional energy sources (gas, oil, hydropower, …) because they can be generated at will (by burning more gas, for example). But solar and wind energies are not under the control of the energy operator (i.e. they are non-operable), because they depend on the weather. Therefore, they must be forecasted with high accuracy. This can be achieved to some extent by accurate weather forecasts. The *Global Forecast System* (GFS, USA) and the *European Centre for Medium-Range Weather Forecasts* (ECMWF) are two of the most important Numerical Weather Prediction models (NWP) for this purpose.

Yet, although NWP's are very good at predicting accurately variables like "100-meter U wind component", related to wind speed, the relation between those variables and the electricity actually produced is not straightforward. Machine Learning models can be used for this task.

In particular, we are going to use meteorological variables forecasted by ECMWF (http://www.ecmwf.int/) as input attributes to a machine learning model that is able to estimate how much energy is going to be produced at the Sotavento experimental wind farm (http://www.sotaventogalicia.com/en).

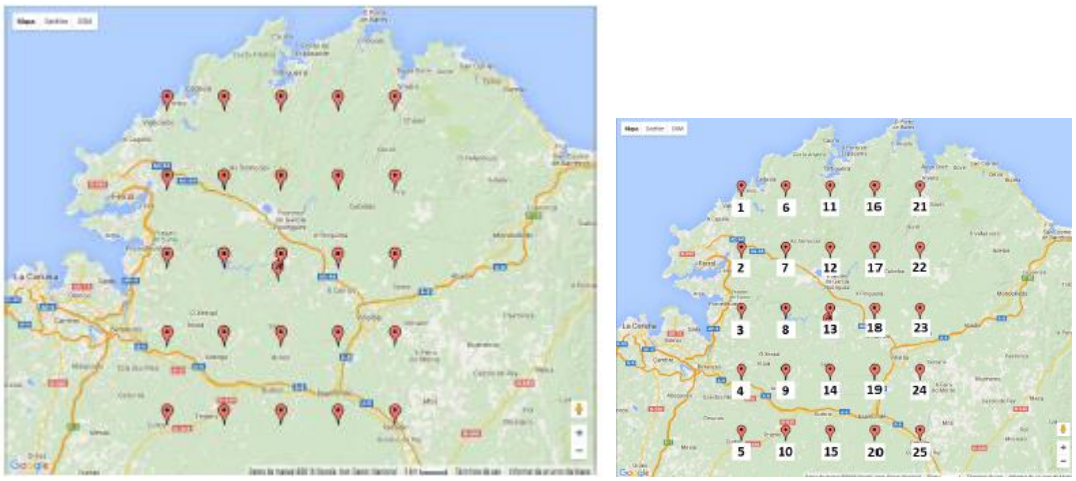
Sotavento wind farm.

More concretely, we intend to train a machine learning model *f*, so that:

- Given the 00:00am ECMWF forecast for variables $A_{6:00}$, $B_{6:00}$, $C_{6:00}$, … at 6:00 am (i.e. six hours in advance)
- $f(A_{6:00}, B_{6:00}, C_{6:00}, \ldots)$ = electricity generated at Sotavento at 6:00

We will assume that we are not experts on wind energy generation (not too far away from the truth, actually). This means we are not sure which meteorological variables are the most relevant, so we will use many of them, and let the machine learning models and attribute selection algorithms select the relevant ones. Specifically, 22 variables will be used. Some of them are clearly related to wind energy

production (like “100 metre U wind component”), others not so clearly (“Leaf area index, high vegetation”). Also, it is common practice to use the value of those variables, not just at the location of interest (Sotavento in this case), but at points in a grid around Sotavento. A 5x5 grid will be used in this case.



5x5 grid around Sotavento.

Therefore, each meteorological variable has been instantiated at 25 different locations (location 13 is actually Sotavento). That is why, for instance, attribute iews appears 25 times in the dataset (*iews.1, iews.2, …, iews.13, …, iews.25*). Therefore, the dataset contains 22*25 = 550 input attributes.

2. WHAT TO DO:

The Sotavento company needs two products from us:

- **(2 points) <u>This corresponds to sections 1 and 2 in the ipython notebook.</u>** The most accurate possible model (and its expected accuracy) using all 550 input attributes. You can choose among KNN, Decision trees, and two ensemble techniques: Random Forests and Gradient Boosting. This is a regression problem, and accuracy will be measured with Mean Absolute Error.
- **(1 point) <u>This corresponds to section 3 in the ipython notebook.</u>** The company wants to know if all 550 input attributes are actually necessary. Is it possible to have an accurate model that uses fewer than 550 variables? How many?

We have historical data available both from ECMWF (for the meteorological variables) and Sotavento (for energy production), from 2005 to 2010. In order to save time, we are not going to use crossvalidation here. For hyper-parameter tuning, we will use 2005-06 data for training and 2007-08 for validation. For model evaluation, data for 2009-10 will be used.

3. WHAT TO HAND IN: All you need to hand in is your ipython notebook. But please, use some of the cells to make comments about what you are doing and your results.