PREDICTING WIND ENERGY PRODUCTION WITH SCIKIT-LEARN (4.0 POINTS)

• 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.

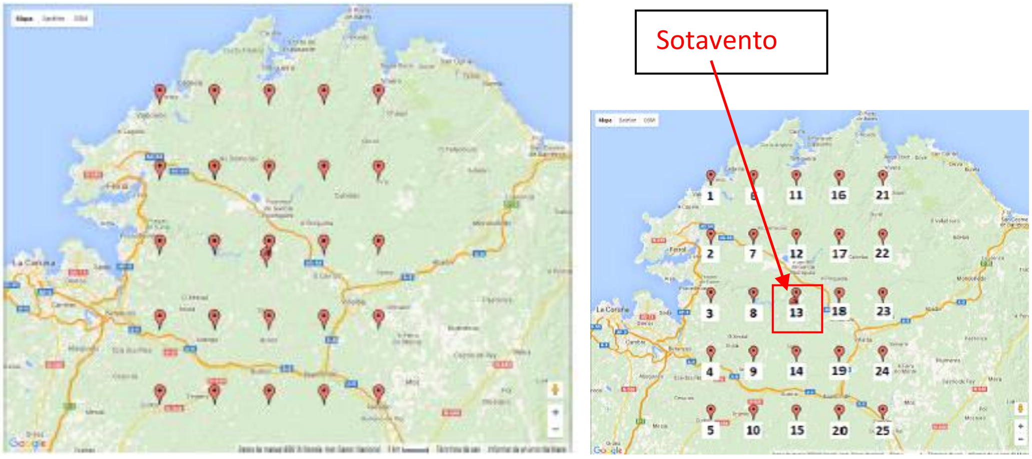
In particular, we are going to use meteorological variables forecasted by ECMWF (http://www.ecmwf.int/) as input attributes to a machine learning model that is able to estimate how much energy is going to be produced at the Sotavento experimental wind farm (http://www.sotaventogalicia.com/en).

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)
- $f(A_{6:00}, B_{6:00}, C_{6:00}, \ldots)$ = energy = 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.

GENERAL CONSIDERATIONS:

1. Results must be reproducible. Therefore, set the seed at the appropriate places. But instead of using seed 42, use the NIA number of one of the members of the group.
2. There are two datasets, the available data set (for model training, hyper-parameter tuning, and model evaluation/estimation of future performance) and the competition dataset (for using the model: making predictions for future instances).

3. Execution time of the training processes should also be reported.

WHAT TO DO:

1. Important: this point is additional to remaining following steps:
    a. (0.5 points) In addition to what you are asked in the following steps, please do something original that you consider relevant.
    b. (0.4 points) You can use ChatGPT for working in this assignment, but I would like you to report (separate file) a summary of how you have used it, some important prompts, some places where ChatGPT went wrong and how you solved it. About two pages.
2. (0.4 points) Explore your data and do a **simplified** EDA, mainly in order to determine how many features and how many instances there are, which variables are categorical / numerical, which features have missing values and how many, whether there are constant columns (that should be removed), and whether it is a regression or classification problem (energy is the response variable). You might want to explore other issues you find interesting, but bear in mind that in this assignment EDA is only 0.4 points.

3. (0.2 points) Split your data into train and test, bearing in mind the type of data (i.i.d. vs. non-i.i.d.?) and that the test partition should be representative of the problem. Decide which metric you are going to use.
4. (0.5 points) Using default hyper-parameters, evaluate Trees and KNN on the testing partition. For KNN, you should a pipeline with preprocessing included (scaling, at least). You can compare 2 scaling methods for KNN.
5. (0.8 points) Now, do hyper-parameter tuning (HPO) for trees and KNN. Report a summary (use a table) of your results so far and draw some conclusions about accuracy results, comparing the evaluation of all alternatives tested.
6. (0.4 points) Using the best method of the ones evaluated previously:
   a. Make an estimation of the error that your model might get at the competition (on the test set).
   b. Final model: using the best method, train the final model and use it to make predictions on the competition dataset. Save both the final model and the competition predictions on files.
7. (0.8 points) Feature selection for KNN: using now the optimal number of neighbors found when doing HPO, use grid-search for selecting the optimal number of features (using SelectKBest and two criteria: f_classif and mutual_info_classif). Is the number of features selected much smaller than the original number? Which are the most important features? Are they related to wind? Are they related to the Sotavento location? Are results improved for KNN compared KNN without feature selection?

WHAT TO HAND IN:

- One or several jupyter notebooks. Please, use some of the cells to make comments about what you are doing and your results. Also, try to justify your decisions. If your group has two members, please write your names at the beginning of the notebook. You can also hand in a file with Python code, and a separate report, if it is more convenient to you.
- A file containing your final model.
- A text file containing the predictions of final model.

- Please, submit just one zip file. Don’t forget to write the names of the two members in the code and the report.

APPENDIX: ATTRIBUTE/FEATURE NAMES:

- t2m: 2 metre temperature
- u10: 10 metre U wind component
- v10: 10 metre V wind component
- u100: 100 metre U wind component
- v100: 100 metre V wind component
- cape: Convective available potential energy
- flsr: Forecast logarithm of surface roughness for heat
- fsr: Forecast surface roughness
- iews: Instantaneous eastward turbulent surface stress
- inss: Instantaneous northward turbulent surface
- lai_hv: Leaf area index, high vegetation
- lai_lv: Leaf area index, low vegetation
- u10n: Neutral wind at 10 m u-component

- v10n: Neutral wind at 10 m v-component
- stl1: Soil temperature level 1
- stl2: Soil temperature level 2
- stl3: Soil temperature level 3
- stl4: Soil temperature level 4
- sp: Surface pressure
- p54.162: Vertical integral of temperature
- p59.162: Vertical integral of divergence of kinetic energy
- p55.162: Vertical integral of water vapour