Quant exercise -- Norlys Energy Trading

One of your colleagues have discovered a trading strategy that looks to be quite profitable. The strategy works by opening a position on the DAH (Day Ahead) market, and then trade this out ID (intraday). It's however starting to decline in performance, so you're tasked with coming up with a new strategy.

The new strategy shall be based on the forecasted wind and solar production as well as the forecasted consumption. The original strategy also contains derived features (an example could be difference in solar level from rolling two-week average). To compare the two strategies directly, you've been given a file (data_for_exercise.csv), that contains the following columns.

data_input.columns = [ts, wind, solar, cons, spot, market]

> ts: Timestamp [-]
> wind, solar, cons: Forecasted wind, solar and consumption [MW].
> spot: The price we're able to open the positions at [EUR/MWh].
> market: The price we're able to close the positions at [EUR/MWh]

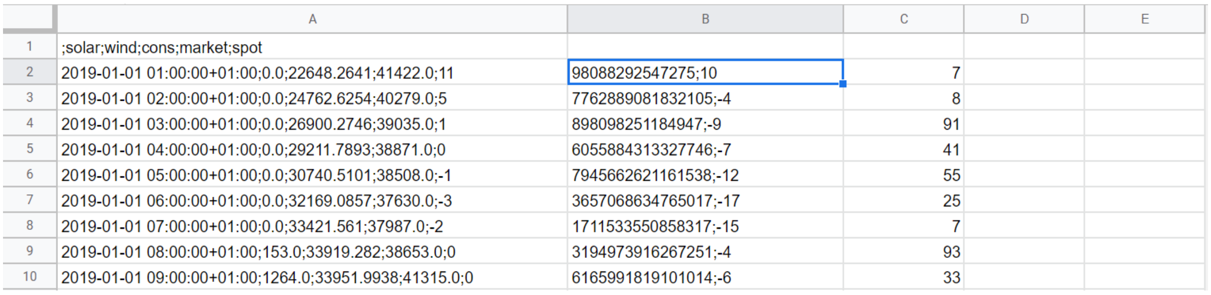
Quant exercise -- progress made by Tom Kent (31.10.21)

Loose plan and strategy

First of all, I am unsure exactly how to define the problem here and devise a 'strategy'. But I shall explore the data, look at some informative relationships between data variables (energy forecasts and prices), and use this initial exploration of the data to inform the construction of some models that predict 'profitable trading'. By this I mean that I focus on when ID price [market] is higher than SAH price [market], since I assume we want to buy low on spot (DAH) and sell high on ID market. I shall use the given energy forecasts and some derived quantities based on moving averages as predictors, in order to predict when market price > spot price. This is therefore a classification problem and I assign binary scores for this. Some more details, helpful plots, and general comments follow below.

1. Inspection of data file -- see <checkdata.py>

- Data is not currently structured for use -- see sample below.
- Delimiter is semicolon [;]. Unclear whether decimals are indicated via comma or point?



	A	B	C	D	E
1	;solar;wind;cons;market;spot				
2	2019-01-01 01:00:00+01:00;0.0;22648.2641;41422.0;11	98088292547275;10	7		
3	2019-01-01 02:00:00+01:00;0.0;24762.6254;40279.0;5	7762889081832105;-4	8		
4	2019-01-01 03:00:00+01:00;0.0;26900.2746;39035.0;1	898098251184947;-9	91		
5	2019-01-01 04:00:00+01:00;0.0;29211.7893;38871.0;0	6055884313327746;-7	41		
6	2019-01-01 05:00:00+01:00;0.0;30740.5101;38508.0;-1	7945662621161538;-12	55		
7	2019-01-01 06:00:00+01:00;0.0;32169.0857;37630.0;-3	3657068634765017;-17	25		
8	2019-01-01 07:00:00+01:00;0.0;33421.561;37987.0;-2	1711533550858317;-15	7		
9	2019-01-01 08:00:00+01:00;153.0;33919.282;38653.0;0	3194973916267251;-4	93		
10	2019-01-01 09:00:00+01:00;1264.0;33951.9938;41315.0;0	6165991819101014;-6	33		

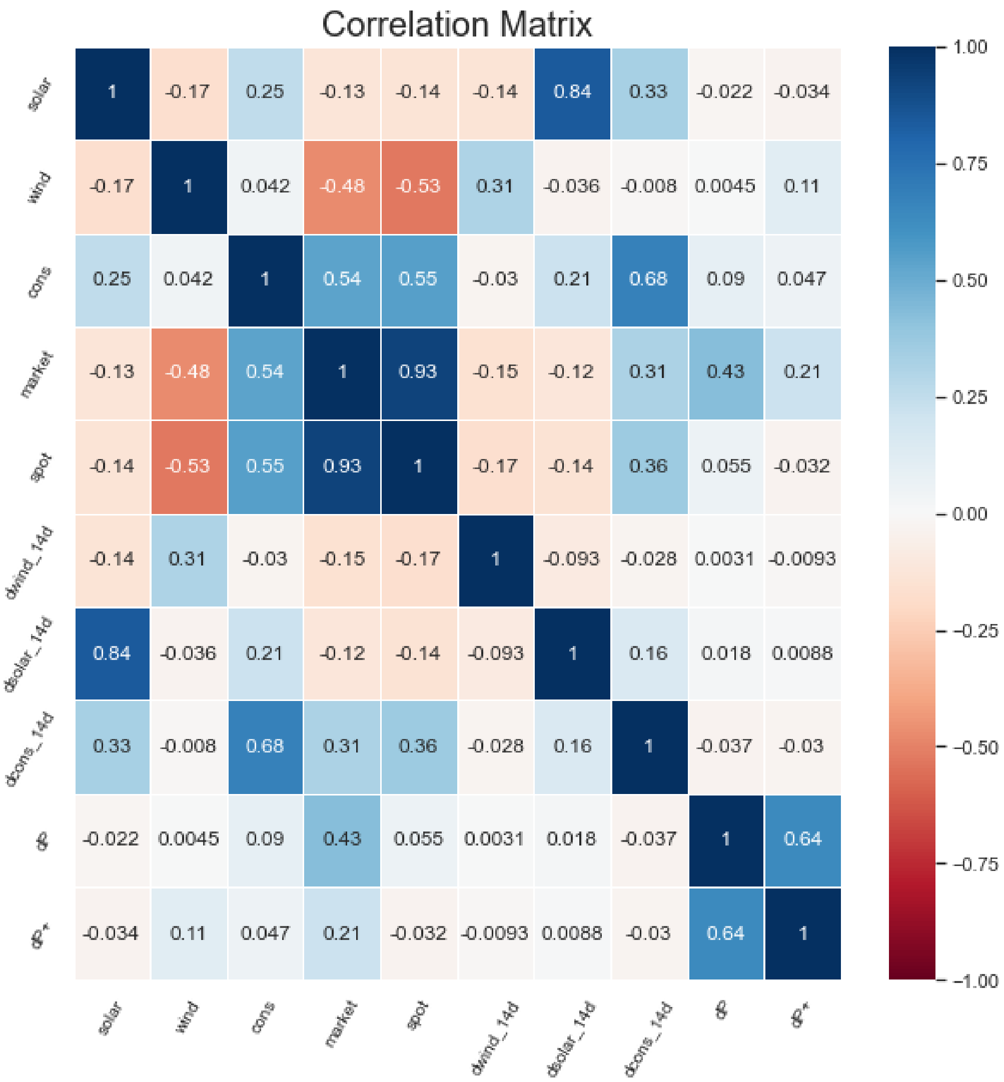
- Working with pandas in Python: <checkdata.py> restructures data_for_exercise.csv into a suitable data frame with the timestamp as index.
- Output: new file <data_new.csv> with amenable structure for analysis.

2. Initial exploration of data via summary stats and plots -- see <exploredata+models.py> (upto line 173)

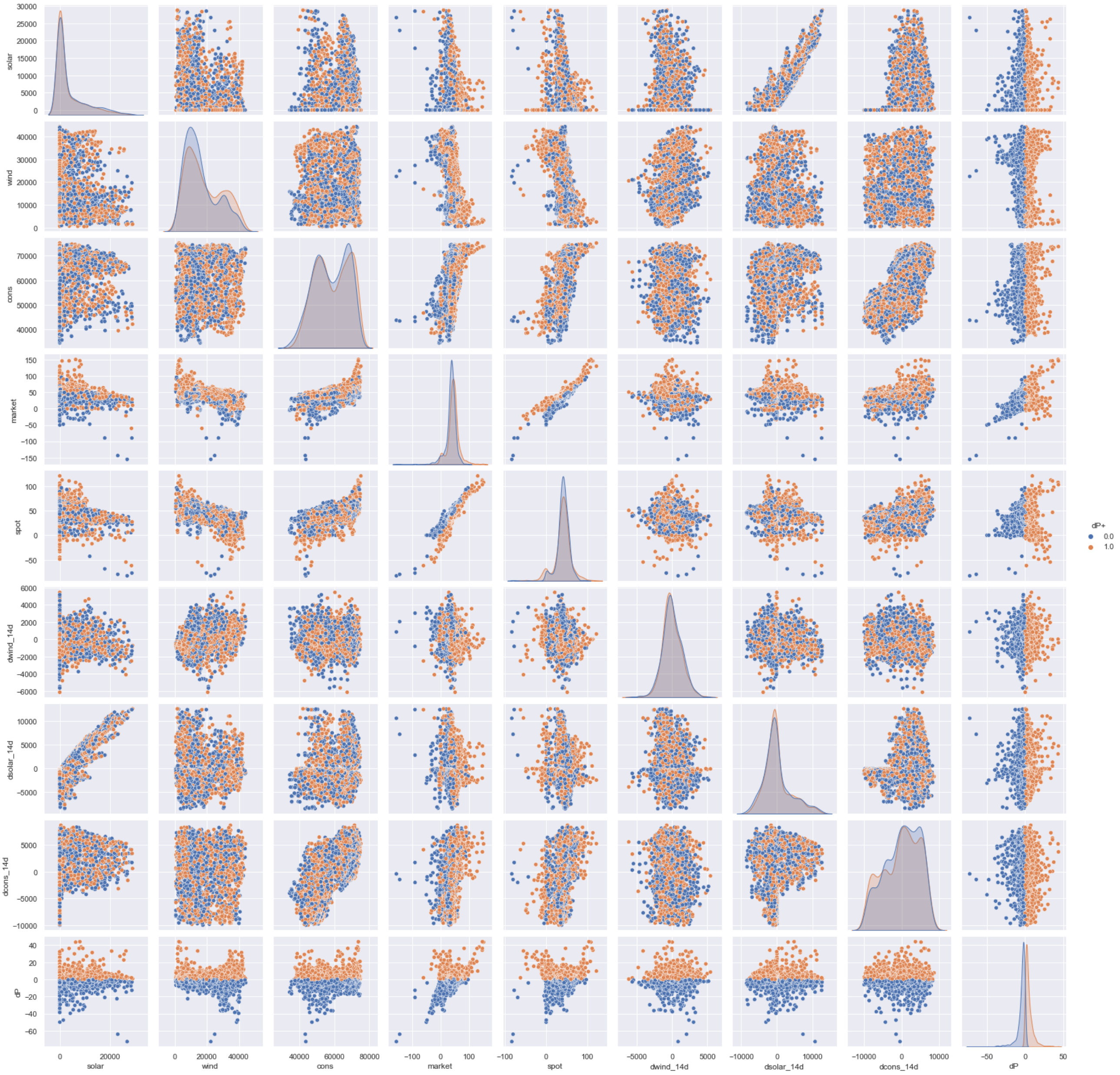
- File uploaded from output of <checkdata.py> and structured as a relevant dataframe with 5 columns ['solar', 'wind', 'cons', 'market', 'spot'].
- Extra derived features are added to the data frame, including:
    - the difference in energy from 2 wk rolling average,
    - the price difference (dP = market - spot, continuous variable, so that dP>0 means sell at market price for positive gain), and
    - a binary classification of this price difference (dP+ = 1 if dP>0 and 0 if dP<0).

  We are therefore interested in the classification dP+ = 1, as this is when ID market price is higher than spot price.

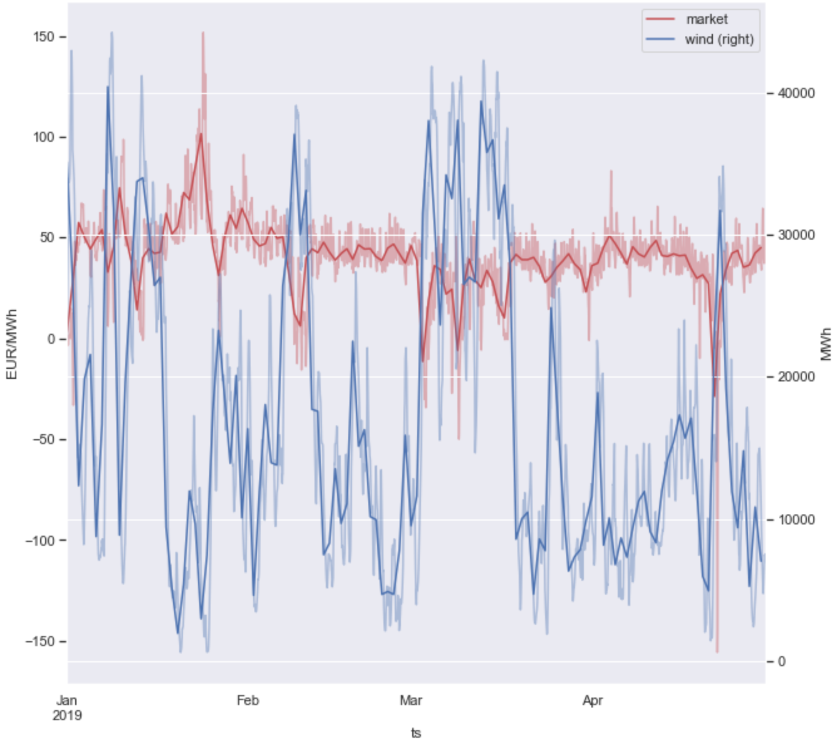
  Initial analysis to inform the relationships between these variables can be plotted. E.g.: scatter plots as a matrix with corresponding correlations. Correlation matrix is shown below:



- Reasonable correlations between solar, wind and consumption and prices. Potential for regression models to predict prices from energy forecasts.
- Some evidence that dP+ can be modelled using energy forecasts and derived features.
- More plots and analysis carried out in an ad hoc manner in the code.
- Another informative plot is a scatter plot with dP+ classified and derived pdfs with respect to this classification (orange is dP+ = 1 -- goal):

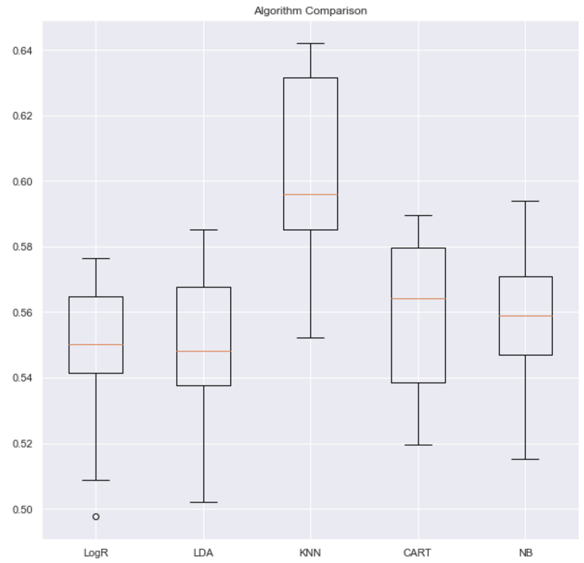


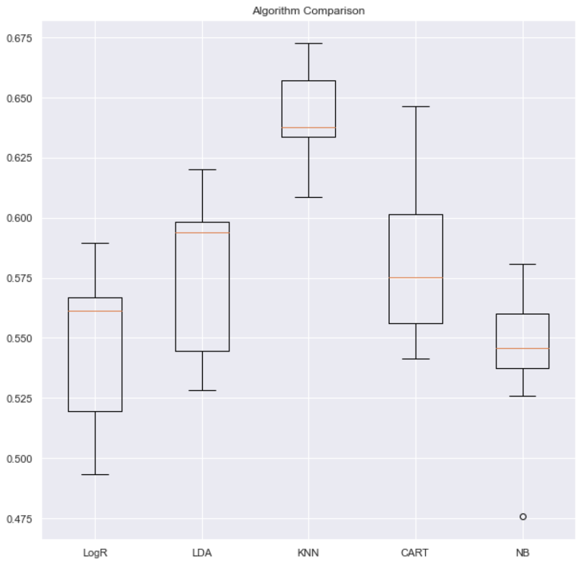
- Here I am particularly interested in the comparison of derived pdfs for wind, which shows a clear separation for the classifications at the upper tail. (High) Wind is perhaps the best indicator of classifying dP+ = 1 (biggest profit).
- Other analysis is carried out in script, including looking at time series. (E.g., wind and market price below. Note peaks in wind correspond with market price decreases.



3. Model building using sklearn libraries -- see <exploredata+models.py> (from line 174 onwards)

- Goal: predict dP+ classification using standard and derived predictors.
- Various standard models from sklearn library will be cross-validated and compared -- note that I am aware of these methods from using two classic datasets (iris and diabetes) for building supervised ML models for classification.
- State predictors and predictand (choices informed by data analysis and plots above)
  - predictors = ['solar', 'wind', 'cons']
  - predictand = ['dP+']
- Cross-validation results (via mean and SD of accuracy from kfold-subsets) outputted and plotted as boxes (below, left). KNN clear winner. Note I played around with number of neighbours to tune this.
- Make predictions: select KNN model and fit using training data. Compare predictions with the test data. Accuracy score: **~61%**. First thoughts: not perfect by any means, but not bad for a first pop and considering the weak correlations above. Confusion matrix shows how many ‘hits’ and ‘misses’ the prediction made: [[208 90] [137 138]]. 208 correct 0s, 138 correct 1s, hence overall accuracy of (208+138)/573 =~0.6.
- Repeat above but now include derived features as predictors:
  - predictors = ['solar', 'wind', 'cons','dsolar_14d', 'dwind_14d', 'dcons_14d']
  - predictand = ['dP+']
- See output of cross-validation (below, right) and compare with first model. KNN clear winner again with an improved accuracy of **~64%.**





4. What does this mean for the strategy?

First iteration of model development has yielded some reasonably accurate predictions for when market prices are higher than spot prices, which could be further tuned for improvement. But I may well be barking up the wrong tree completely!