Electronic Supplementary Material (ESI) for Reaction Chemistry & Engineering.

Industrial Data Science – A Review of Machine Learning Applications for Chemical and Process Industries

Max Mowbray[a], Mattia Vallerio[b], Carlos Perez-Galvan[b], Dongda Zhang[a], Antonio Del Rio Chanona[c] and Francisco J. Navarro-Brull[b,c*]

[a] The University of Manchester, Manchester, M13 9PL, UK
[b] Solvay SA, Rue de Ransbeek 310, 1120, Brussels, Belgium
[c] Imperial College London, London, SW7 2AZ, UK
*E-mail: francisco.navarro@solvay.com

Annex I

Process understanding

Problem definition

The engineering team wants to quickly understand which recorded process variables (tags) can explain the yield variability of a column seen over the years (Figure 1a). The challenge is that these tags show colinearity among them (Figure 1b and c) and there is not a clear linear or seasonal explanation that can explain the yield obtained. As the sample data was carefully taken and cleaned when the process showed steady-state conditions, thus data points can be assumed independent from each other. This can be verified by looking at the autocorrelation of the yield (Figure 1d), but more importantly means that the process dynamics are, in principle, not affecting the results. Likewise, no major changes have occurred both in the process or the control during this period.

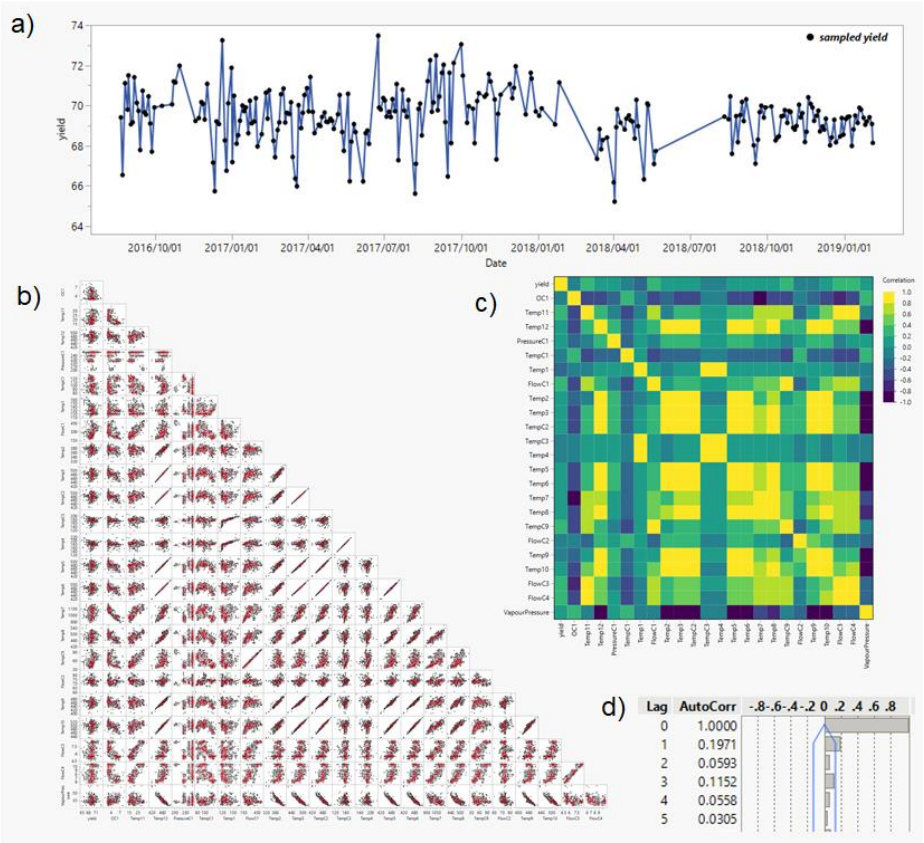

Figure 1. The production yield of a distillation column over the years (a), sampled every 2 to 3 days during its steady-state regime. Scatter plot matrix (b) showing the data densities amongst additional sensor time measurements (tags). Colormap shows pairwise correlations (c), where yellow and dark blue indicates strong colinearity among these tags. The lack of autocorrelation (d) on the target (yield), indicates that process dynamics (delays, lags, and other seasonal behavior) can be disregarded for this preliminary analysis.

Process data

From process knowledge, a broad selection of process variables (tags) has been already extracted. The major challenge at this step is to iterate through data cleaning to perform the first selection of relevant variables. As the yield (target variable) does not show shutdowns or autocorrelation, a screening model can be directly used to ease this prior tag selection. First, a balanced data partition taking into account the yield distribution needs to be done (Figure 2a). The training data set will be used to fit the model, the validation will avoid overfitting and a third independent test data will be solely used to verify what would be the model performance if it went to production (unseen data by the model). For this first screening, simpler models can be used via random selection of the tags and sampled times. The different units or outliers to be present in the data set can be handled with partition (or decision) tree models, which split data to better explain the yield (on average) by looking at the tags independently in each iteration. As

the model grows in complexity (higher number of splits or depth), a better fit is obtained by combining at the end all models that, in principle, capture the features in each data set sampling. This technique is known as bootstrap or random forest (Figure 2c), and while being widely used tends to capture the noise seen in the data (i.e. overfit). To perform a simple variable selection, the contribution of each tag is shown in descending order (Figure 2b) with a cutoff on the known noise which was added into the model as an independent and additional term.

Term	Number of Splits	SS	Portion
FlowC1	29	17.1678444	0.1988
TempC9	27	12.2012469	0.1413
Temp1	13	9.75679814	0.1130
Temp4	18	5.57623969	0.0646
TempC3	13	4.64860694	0.0538
OC1	15	3.66068252	0.0424
FlowC2	22	3.53264433	0.0409
PressureC1	9	2.42725587	0.0281
TempC1	14	2.29403566	0.0266
Temp6	9	2.23790975	0.0259
Shuffle[yield]	14	2.17379588	0.0252
Random Normal Noise	11	1.95953512	0.0227
VapourPressure	10	1.94971732	0.0226
FlowC3	11	1.89361028	0.0219
Temp12	12	1.889938	0.0219
Temp3	11	1.85201822	0.0214
Temp9	11	1.82578396	0.0211
FlowC4	8	1.76030855	0.0204
Temp2	8	1.52285823	0.0176
Temp10	12	1.44849998	0.0168
Temp5	7	1.12753413	0.0131
Temp7	8	0.91416155	0.0106
Temp11	7	0.80503604	0.0093
Random Uniform Noise	6	0.75906398	0.0088
Temp8	11	0.69334372	0.0080
TempC2	8	0.28127547	0.0033

Target	yield	Training Rows:	127
Validation Column:	Validation stratified	Validation Rows:	76
		Test Rows:	50
Number of Trees in the Forest:	23	Number of Terms:	26
Number of Terms Sampled per Split:	6	Bootstrap Samples:	101
		Minimum Splits per Tree:	10
		Minimum Size Split:	5

	RSquare	RASE	N
Training	0.622	0.8338901	127
Validation	0.276	1.1315936	76
Test	0.287	1.1229573	50

Figure 2. A stratified random sample of the yield (a) maintains equally the data distributions of the training, validation, and test data sets. This split can be done if data is selected when steady-state conditions are reached in the process. Synthetic noise variables as factors (shuffled, random, and uniform noise), the first selection of tags can be done looking at the top column contributions (b) of a random forest model (c).

Process modeling:

With the subset of variables above the noise obtained previously, one can proceed to visualize and interpret the data. A single partition tree (Figure 3a) captures the strongest factors with a simple structure (Figure 3c). Yield nonlinearities varying the flow and temperature appear if a kernel density is used to smooth the response (Figure 3b). For the tree, a random k-fold validation split was used to better utilize all the data (5 models built using 20% of random samples for validation). Yet, to avoid overfitting and keep relevant tags only, the 40 times split tree model with higher R2 (Figure 3d) was pruned until the noise factors disappeared (Figure 3f). This tree model captures the nonlinearities in a robust manner and with a simple logic (Figure 3e) that can be shared or implemented in any control logic.

At this stage, it is important to go back to the problem definition before continuing. By looking at the yield again (Figure 4a), one can appreciate certain drops in the yield. While these seemed outliers, there is one assumption that is hardly always met: the total independence between consecutive data samples (e.g. process dynamics). By introducing, for example, a first-order differentiation of the tags as new factors; the difference of pressure explains the sudden yield drops (Figure 4a and the model shown in 4b). Notice that local changes can be overlooked by the model performance indicators as models tend to find what explains the (average) correlations between tags over the years and not the trends (dynamics). A typical R2 curve of training/validation/test when increasing model complexity is shown in Figure 4c. Without proper cleaning of data to have only steady-state information, time-series split and model inputs capturing dynamic modeling will be needed (the reader is referred to in the main manuscript).

Until this point, the methodology of variable selection here described can be easily parallelized and therefore does not limit engineers to creating all the non-linear combinations that can be usually seen in chemical processes. A library built to this end is called ALAMO, for example. Yet, the use of Neural Networks (NN) as general functions approximations can empirically capture higher degrees of nonlinearities among tags. To illustrate this, a fitted NN output is compared to the ground truth (Figure 5a). The yield was a Rosenbrock function that was calculated to test this methodology using the captured flow and temperature. Additional random and uniform noise was added as well as the effect of the identified pressure changes. A two-layer feed-forward NN (Figure b) can fit these complex data interactions (Figure 5c, d, and e), once the important tags are given as inputs.

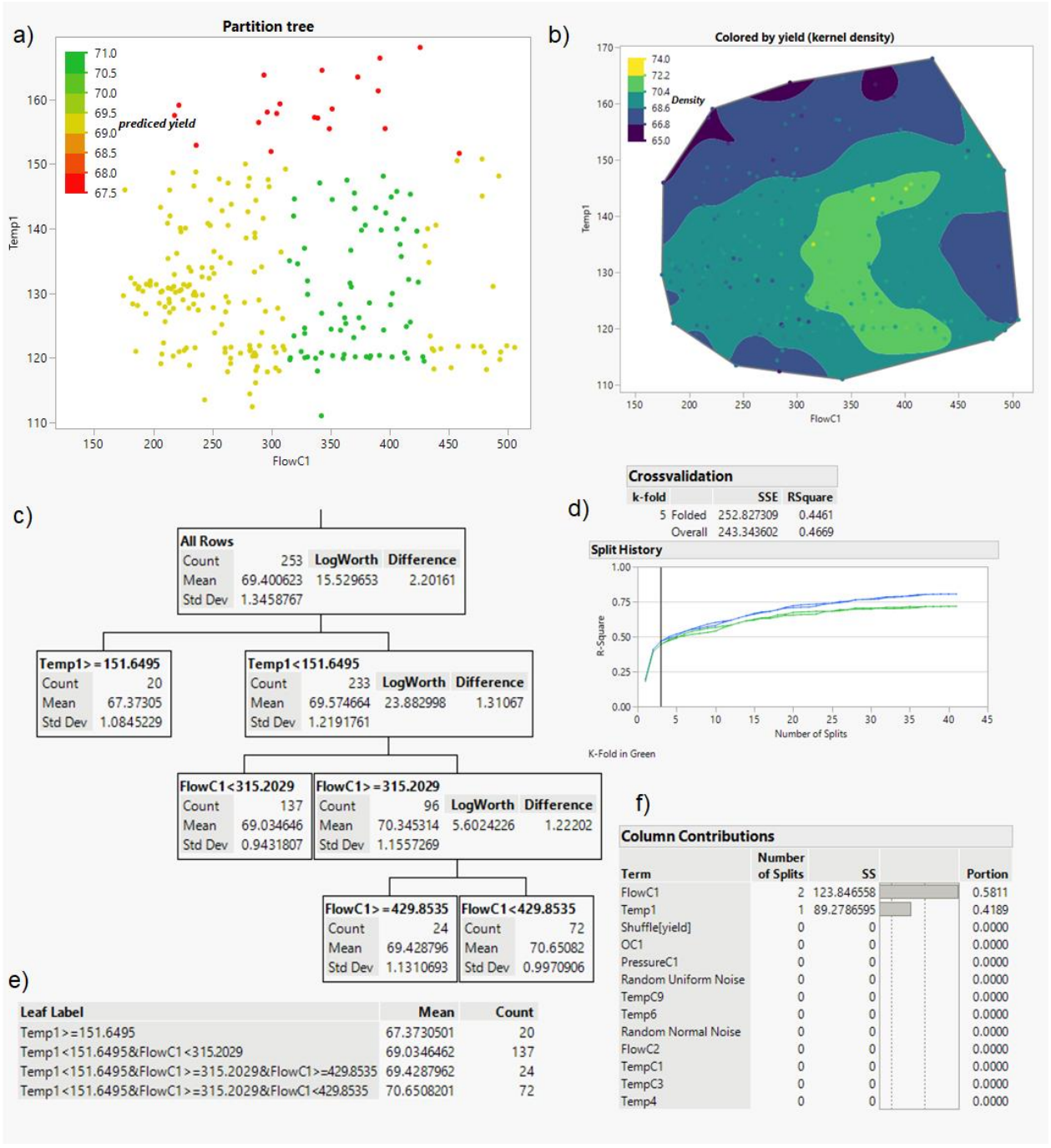

Figure 3. A decision tree model is used to create partitions of data (a) in several regions that capture the yield nonlinearities more simply compared to a kernel density contour plot (b). The tree (c) provides an interpretable robust logic that captures the variability, in average (e), using a simple k-fold validation technique (d). While lower error can be mistakenly achieved (d) by increasing the number of splits, the appearance of random variables and their pruning is used to avoid overfitting and keep this model easy enough for interpretation.

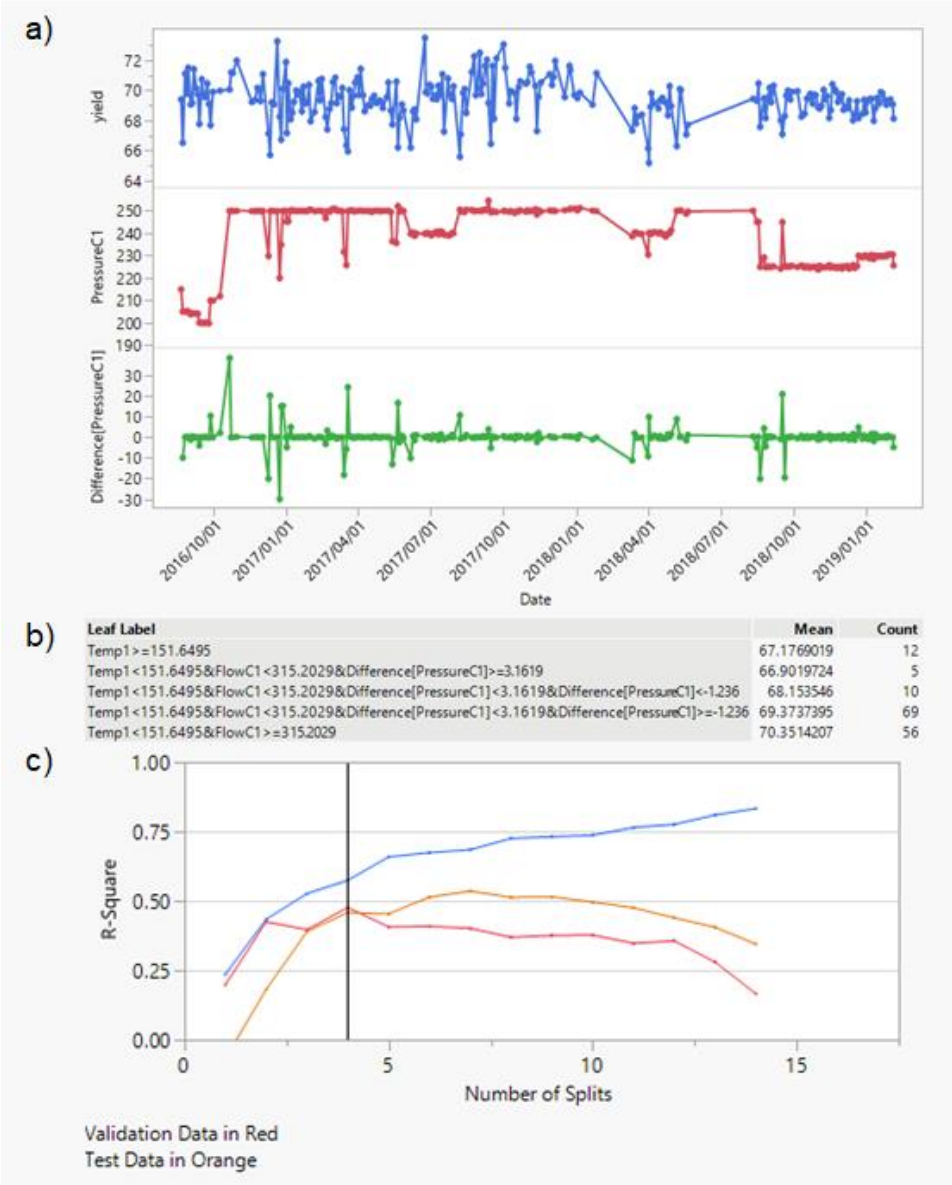

Figure 4. The creation and screening of additional features in a second iteration identified how changes in pressure between consecutive measurements were affecting the yield in the short term (a). This partition tree model (b) using the training/validation/test data split (c) obtains better R2 scores while avoiding overfitting.

As illustrated, data-driven techniques can be used to quickly capture and understand complex interactions amongst imperfect industrial data sets. It is important to emphasize that pure empirical models are well suited for data interpolation. Additional data points following a design of experiments technique and/or the combination with first principles are recommended if the variability is not enough and the optimal values are found towards an unseen interval (e.g. risk of extrapolation). This industrial data set can be downloaded and further adapted to test the robustness of the methodology here described.

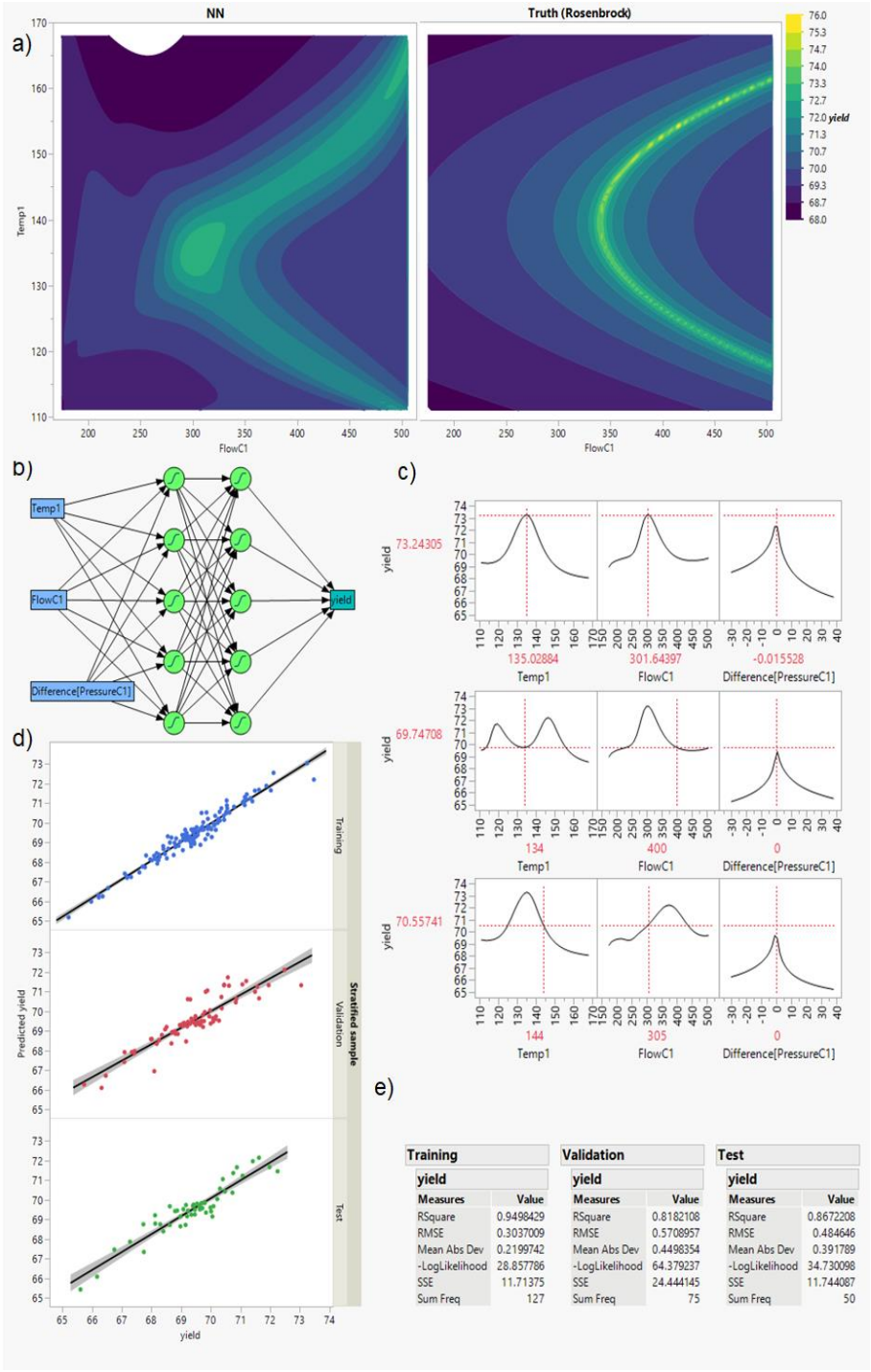

Figure 5. Comparison between the empirical model learned by a neural network and the ground truth (a) which was manually conditioned by pressure changes and additional random and uniform noise. Two layers with five nodes each are used to build a feed-forward neural network with hyperbolic tangents as activation functions (b). The flow, temperature, and pressure non-linear effects of each variable (c) are well captured for the training/validation/test data (d and e).