JMP Academic Case Study 067

Chemical Process Improvement in Resin Production: Part 2

Online process monitoring, spectral data Modeling, machine learning

Produced by

Frank Deruyck, HOGENT University of Applied Sciences
frank.deruyck@hogent.be

Volker Kraft, JMP Global Academic Team
volker.kraft@jmp.com

jmp STATISTICAL DISCOVERY

Chemical Process Improvement in Resin Production: Part 2

Online process monitoring, spectral data modeling, machine learning

Key ideas

A chemical company producing resins faces two problems in production: On the one hand, the quality of the produced resins is too low. On the other hand, the required offline measurements to detect defects is very inefficient. In this second of two case studies, we implement online measurement and analysis of spectral data for enhanced process monitoring and efficiency.

Background

As outlined in the first case study, BLX Chemicals, a leading producer of high-quality resin, made significant progress improving both the process stability and capability. Using a process model, better operating conditions (process settings) could be implemented leading to better purity, which is considered as the critical quality attribute.

The primary objective of this second project is the elimination of the offline purity measurement, leading to a more efficient production process. As a result, we hope to replace the offline purity monitoring step with online monitoring based on spectral (NIR) data. Near-infrared (NIR) analysis is a fast, nondestructive technique that measures how a material absorbs light in the near-infrared region to determine its chemical and physical properties. In offline measurements, samples are taken to a laboratory instrument for a detailed analysis. In online measurements, NIR sensors are integrated directly into the production process, enabling continuous, real-time monitoring and rapid adjustments and making NIR a powerful tool for improving quality, efficiency, and process control.

The task

During this study, we investigate if the spectral data can help to predict process quality, i.e. a purity above 90% (the lower spec limit). We use NIR spectra together with their purity measurements from the initial experiment (24 samples), as well as 40 additional samples from the field (production data). Different methods summarizing the spectra are discussed, as well as different methods for modeling and machine learning.

These are the suggested analysis steps:

1. Explore the spectral NIR data.
2. Build and assess purity-NIR models:
    - DOE data: Build a first model (PLS).
    - With more data: Explore modeling options and select a purity-NIR model.

The data

I-optimal DOE and spectra.jmp

Run #	Experiment run #
Random Block	Blocking factor for day (1, 2, 3, 4)
Temperature	Continuous factor #1 (135-145 °C)
Time	Continuous factor #2 (10-20 min)
Pressure	Continuous factor #3 (100-160 10^3Pa)
Flowrate	Continuous factor #4 (100-180 l/h)
Purity	Response variable with spec limit
Intensities by Wavelength	Column group with spectral data (intensities at 286 wavelengths)

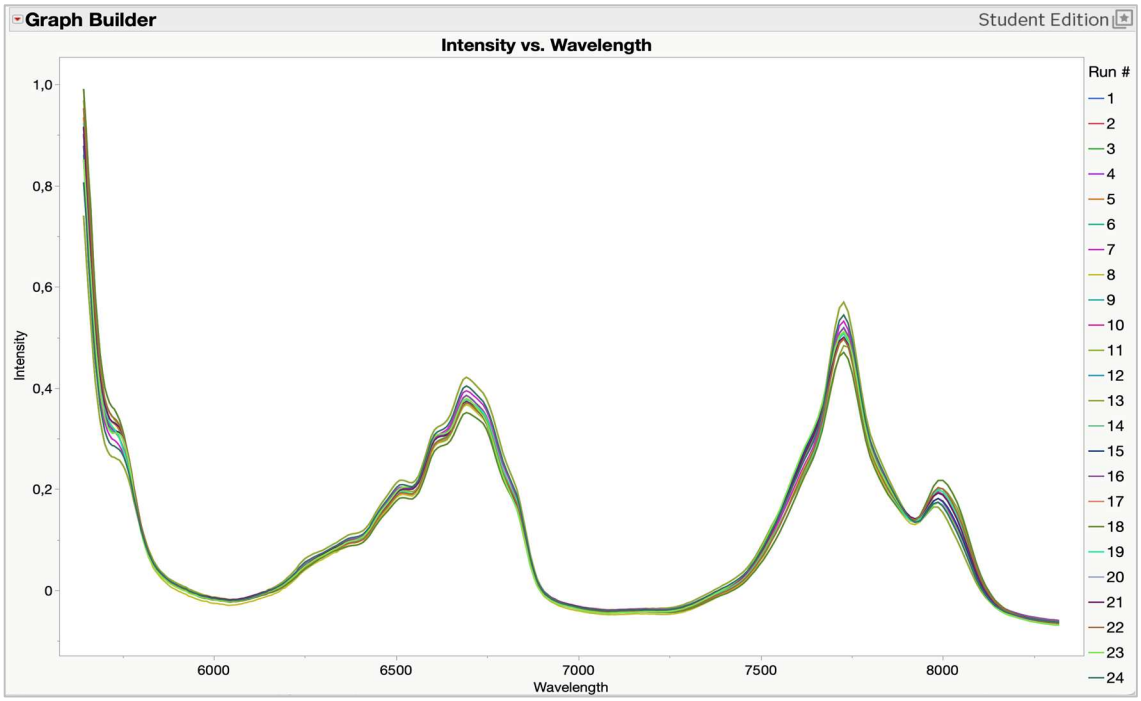
DOE and Prod spectra.jmp

Study	Identification of the two subsets (DOE, Production)
Sample #	Sample ID
Purity	Response variable with spec limit
Intensities by Wavelength	Column group with spectral data (intensities at 286 wavelengths)
X Scores for Prediction Formula	PLS scores for three latent factors

Analysis

Explore the spectral NIR data

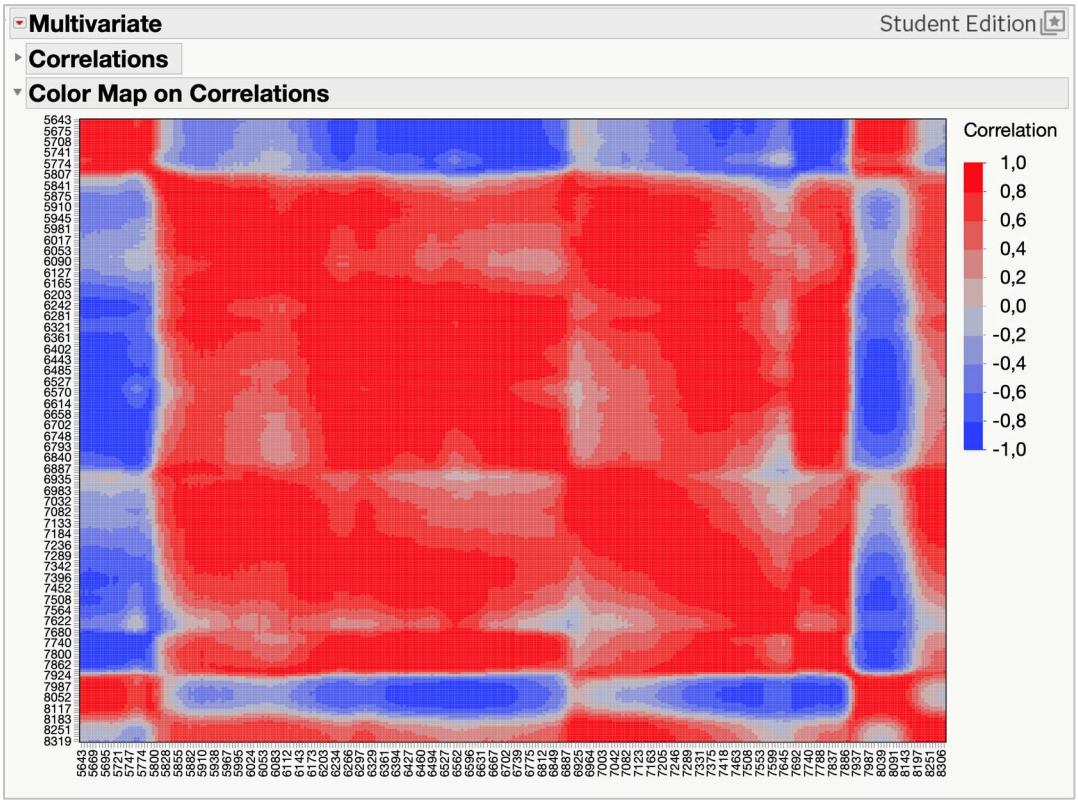
Exhibit 1 Plot of 24 NIR spectra



(Tables > Transpose... Alternatively, run table script Transposed table for Graph Builder > in new table, run table script Intensity vs. Wavelength to create this plot.)

Before we launch any analysis, let us visually explore the new data we have in the first data se. In addition to the process data we already saw in Part 1, the Intensities by Wavelength column group represents one spectrum per row, measured at 286 wavelengths set as the column name. Each cell value represents an intensity measurement at the given wavelength. Plotting the spectra in Graph Builder requires some rearrangement of the table structure, which can be achieved using the Transpose platform. For convenience, a table script can be used to recreate a transformed data table, which has a script to plot all spectra in Graph Builder (Exhibit 1).

Exhibit 2 Correlation map of spectra data



(Analyze > Multivariate Methods > Multivariate, all intensity columns into Y, Columns > OK.)

The intensity plot looks very typical for spectral data, with some good news supporting the next steps: While all spectra seem to follow a similar pattern overall, some ranges show subtle differences between the individual spectra. Our task is to peel out these differences and systematically relate them to process quality, in this instance, purity. Other good news is that all spectra seem to be well aligned with interesting parts spread over the whole x range and don't suggest applying any data clean up or preprocessing up front.

A warning, though, becomes very obvious from the correlation analysis (Exhibit 2). As expected, the intensity vectors at adjacent wavelengths are highly correlated (positively in red or negatively in blue). It makes the existing data inappropriate for regression modeling or for any other modeling technique that is not robust against collinearity. However, since we have to reduce the dimensionality anyway during the next analysis step, the correlation should be eliminated as well.

Building a partial least squares model

In light of the obvious collinearity and high dimensionality of the spectral data, the analysis team suggested to run a principal component analysis (PCA). While PCA is a proven method converting many dimensions (even many more than the number observations!) into a few uncorrelated dimensions (the so-called components) pointing into the direction of maximum variation in our data, PCA is a non-supervised machine learning method, which means that the conversion would not be guided by our response, Purity, (i.e., the principal components) would not reflect any pattern relevant to what we observe on our outcome.

Another method, partial least squares (PLS), is the supervised technique which does exactly that. It can reveal a latent structure in X and the relationship between X and Y. PLS is one of the multivariate analysis platforms in JMP, with built-in tools for model selection, diagnostics, and interpretation. The team selected the PLS with only three factors (Exhibit 3), which explain more than 96% of the variation in the X space and more than 86% of the variation in the Y space. Note that the conversion in X replaces the 286 original intensity columns by only three X score columns – a massive data reduction. In addition, it can be easily proven (using the Multivariate platform) that the three X scores are perfectly uncorrelated.

Exhibit 3 Partial least squares (PLS) analysis for three factors

Partial Least Squares Student Edition

Y response: Purity

Model Launch

Model Comparison Summary

Method	SVD	Number of rows	Number of factors	Percent Variation Explained for Cumulative X	Percent Variation Explained for Cumulative Y	Number of VIP > 0.8
NIPALS	Fast	24	3	96,248908	86,523646	174
NIPALS	Fast	24	2	75,831912	85,671113	174

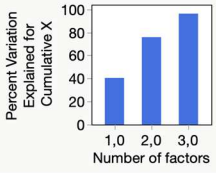
KFold Cross Validation with K=7 and Method=NIPALS Using Fast SVD

NIPALS Fit with 3 Factors Using Fast SVD

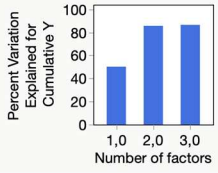
X-Y Scores Plots

Percent Variation Explained

Number of factors	X Effects	20406080	Cumulative X
1	40,4933		40,493
2	35,3386		75,832
3	20,4170		96,249



Y Responses	20 40 60 80	Cumulative Y
50,2331		50,2331
35,4381		85,6711
0,8525		86,5236

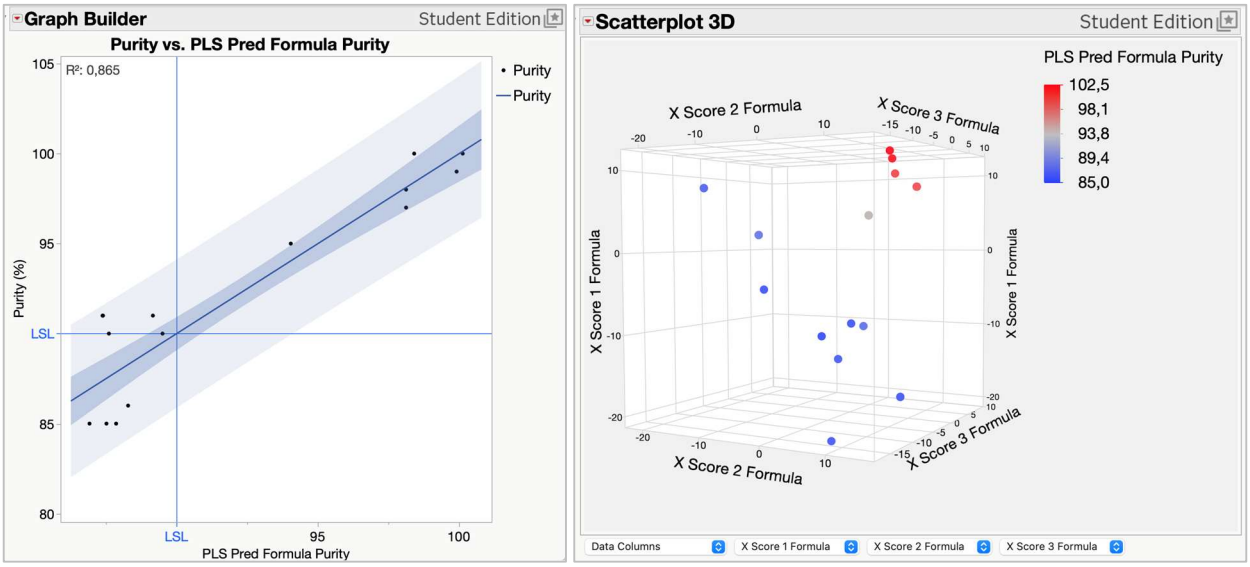


(Analyze > Multivariate Methods > Partial Least Squares, Purity > Y, Response; all intensity columns into X, Factor > OK > Go. Create a NIFALS Fit with three factors, then Save Columns > Save Prediction as X Score Formula.)

The PLS model performance can be assessed in different ways. The left graph of Exhibit 4 shows an Actual Purity by Predicted Purity plot. While the overall performance seems to be reasonable, and all data points fall within the prediction confidence band, there are two false alarms (actual purity above LSL but predicted purity below LSL) and two data points on the edge (predicted purity as too low but actual purity close to LSL).

The coverage of the X space by our 24 experimental runs can be explored by creating a Scatterplot 3D (Exhibit 4, right graph). Exploring this space from different angles by turning the 3D graph around reveals that a significant portion of the X space remains unexplored (“white space” without any close data points). This observation led to the suggestion to enhance the data set by more spectral measurements from the production site. Luckily, 40 more production spectra could be measured and added to the data set. This enhanced data set is used throughout the second part of this exercise.

Exhibit 4 PLS model assessment



(Left: Graph > Graph Builder, Purity > Y axis, Pred Formula Purity > X axis, add Confidence for Prediction and Statistics R^2. Right: Graph > Scatterplot 3D, X Score formula 1,2,3 > Y, Columns, Pred formula Purity > Coloring.)

Explore modeling options and select a purity-NIR model

Motivated by the initial partial least squares (PLS) model, the quality team discussed several directions enhancing the purity-NIR model. First, 40 additional spectra with Purity measurements have been added to the 24 original runs from the experiment, yielding a total of 64 spectra. Second, alternative data reduction techniques have been considered replacing PLS scores. While principal component analysis (PCA) was given up as an unsupervised alternative in favor of PLS, functional data analysis yielding functional principal components (FPC) seems to have great support, especially for spectral data (and can be explored on your own during an exercise).

As a third dimension, there was some discussion on how to explore other modeling techniques to replace the PLS model. While PLS can be seen as a standard multivariate method for problems in such fields as chemistry or material sciences, JMP's modeling toolbox has many more options for predictive modeling and machine learning. Instead of building and comparing multiple model candidates manually one by one, the use of the Model Screening platform in JMP has been suggested for a first half-automated inspection of many models. A cross-validated ranking of machine learning models predicting Purity is shown in Exhibit 5. The most promising models are neural boosted, K nearest neighbors and support vector machines.

The fourth topic that the team discussed was how to protect a model from overfitting. A perfect model does predict just the systemic pattern (the underlying signal) of a system or process but not any noise in the training data. Cross validation selects the best model based on validation data, while the models are built using training data. Some studies even put so-called test data aside for future honest model assessment and comparison using so far unseen data. JMP's support for model validation includes setting a random holdout portion, adding a validation column splitting all rows into training/validation/(test) subsets, or K-fold cross validation. Model screening can be configured to use K = 3 folds, with each fold being used for validation once, for training twice, and then averaging all results.

For further finetuning of a model, a deep dive into model diagnostics, or to save prediction formulas of interest, select a method and click Run Selected.

Exhibit 5 Model screening result

Model Screening for Purity Student Edition

Table: DOE and Prod spectra.jmp **Response:** Purity

Details

Summary Across the Folds

Method		N Trials Folds	Sum Freq	Validation Set Folds: Mean RSquare	StdDev RSquare		Mean RASE	StdDev RASE	
Neural Boosted	†	3	21,3333	0,8969	0,02022		1,5237	0,34196	
K Nearest Neighbors	†	3	21,3333	0,8227	0,05511		1,9887	0,53283	
Support Vector Machines		3	21,3333	0,7626	0,11019		2,3089	0,83984	
Generalized Regression Lasso	†	3	21,3333	0,7472	0,04081		2,3666	0,37661	
Fit Stepwise	†	3	21,3333	0,7471	0,04090		2,3663	0,37128	
Partial Least Squares	†	3	21,3333	0,7456	0,04229		2,3726	0,37043	
XGBoost		3	21,3333	0,7155	0,03866		2,5060	0,30002	
Fit Least Squares		3	21,3333	0,7149	0,08459		2,4800	0,30103	
Bootstrap Forest	†	3	21,3333	0,7085	0,08079		2,5558	0,65964	
Boosted Tree	†	3	21,3333	0,6849	0,21769		2,4781	0,46501	
Decision Tree	†	3	21,3333	0,5129	0,24210		3,1732	0,53667	

† indicates that the method uses validation data to tune model.

Select Dominant Run Selected Save Script Selected

Training

Validation

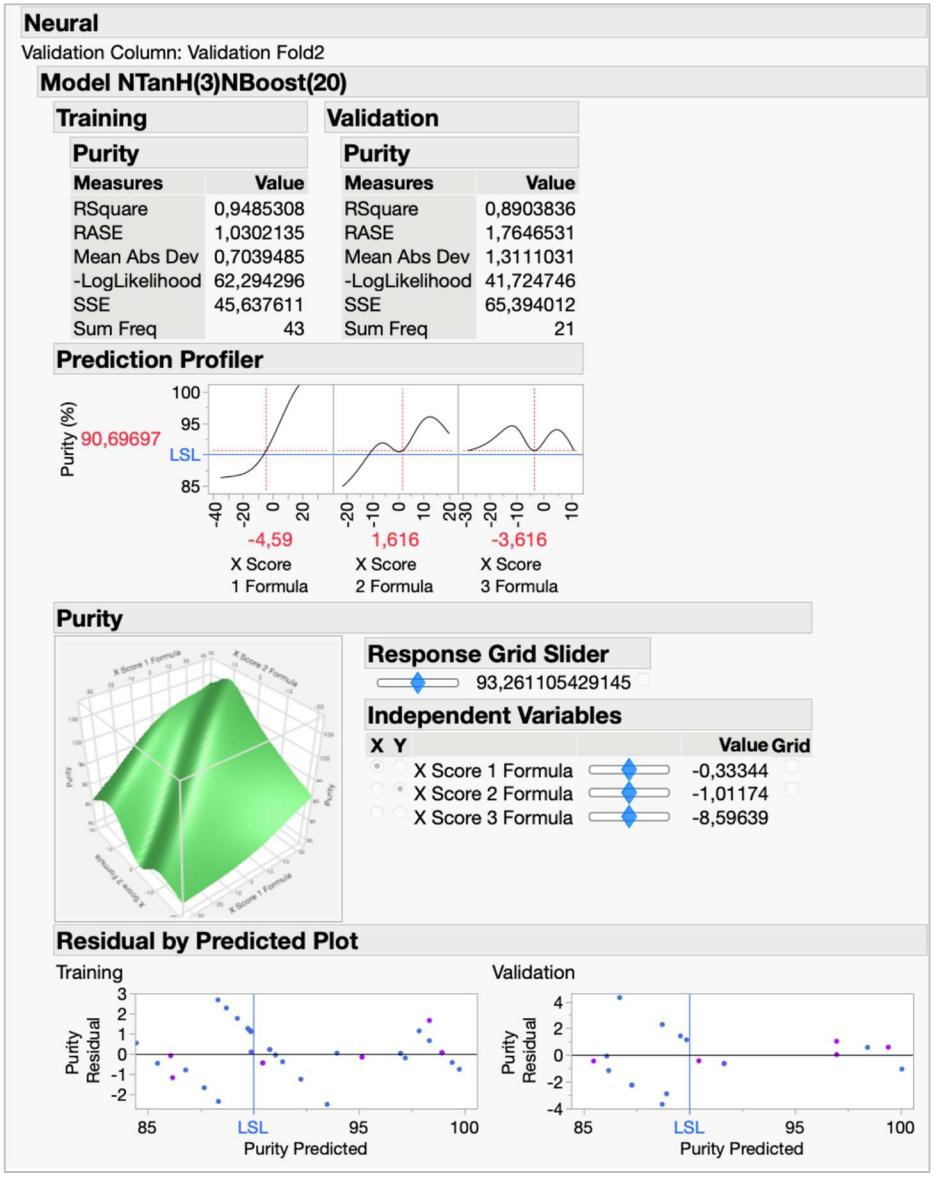
Test

Sum Freq and Sum Weight are suppressed when they are the same as N.

(Analyze > Predictive Modeling > Model Screening, Purity > Y, Response, X Score formula 1-3 > X, Factor. Select all methods of interest, Random Seed = 54321, Cross validation K=3. Use two-way splits for K fold; no third test set needed!)

Exhibit 6 shows some model output after selecting and running the “best” screened model, a boosted neural network. Despite many other fine-tuning options building a neural network, the mean validation R-square is already reported as 0.87 – a great result, especially in the light of the massive data reduction carried out on the NIR spectra. Another result supporting the neural network model is its mean RASE, which is the smallest for all models considered. Since neural networks tend to overfit, the application of some validation cannot be highlighted too much.

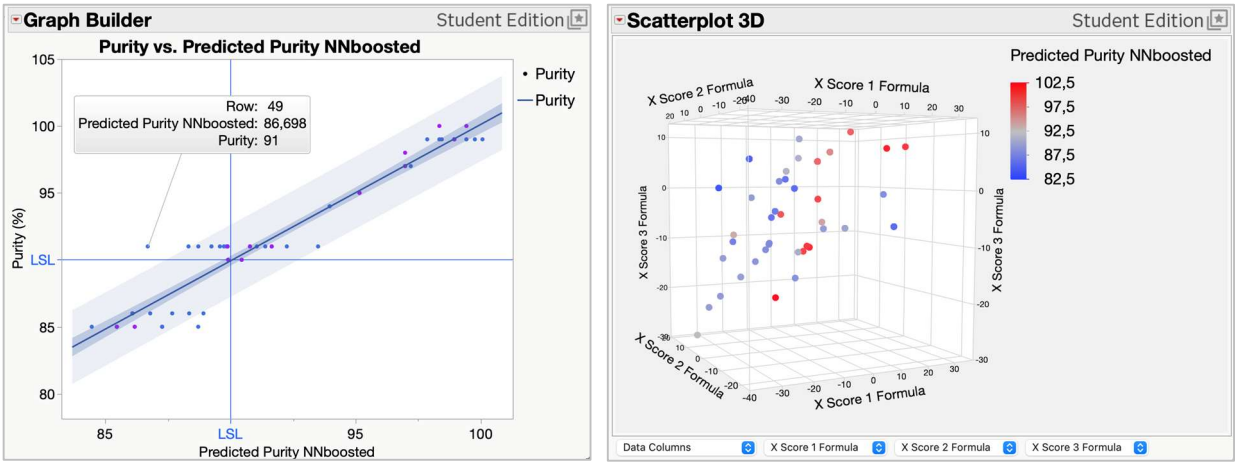
Exhibit 6 Boosted neural network



(Left: Graph > Graph Builder, Purity > Y axis, Pred Formula Purity > X axis, add Confidence for Prediction and Statistics R^2. Right: Graph > Scatterplot 3D, X Score formula 1,2,3 > Y, Columns, Pred formula Purity > Coloring.)

However, even this “best model” is not all perfect. The Residual by Predicted Plot shows some non-random pattern in the lower range of Purity, together with a non-constant error variance.

Exhibit 7 Boosted neural network assessment



(Left: Graph > Graph Builder, Purity > Y axis, Pred Formula NN Purity > X axis, add Confidence for Prediction and Statistics R^2. Right: Graph > Scatterplot 3D, X Score formula 1,2,3 > Y, Columns, Pred formula Purity > Coloring.)

Looking at Exhibit 7, we still see a few falsely classified outputs, with Purity measured above LSL but predicted as out of spec. While some more white space in the X space could be covered by adding more spectra, some unexplored white space remains. Although the best model so far would not ensure an error-free online monitoring of Purity, this study created sufficient confidence that a fully automated online monitoring would be feasible.

Summary

Statistical insights

While the spectral (NIR) data created an opportunity to improve the process efficiency, using so many highly correlated variables requires dedicated multivariate methods for data transformation. Options discussed in this study included PLS, PCA, and FDA.

Model screening has been applied as an efficient way finding promising modeling methods for the problem at hand. Using model screening in this study, the team was pointed to several promising machine learning methods (including some that the team had never used before, so they were not on their radar from the start).

Building predictive models, we also discussed the need to protect against overfitting and the use of diagnostics for model assessment and selection.

Implications and further study

While the chosen modeling method leaves some (minor?) opportunities for further fine-tuning, enhancing the model by more (and better!) data has been set as a prioritized direction for further studies. Some ideas include:

- Investigate and reconfirm the suspicious Purity measurements around 91%.
- Use a more intelligent split into training and validation data, e.g., by selecting a subset based on covariate factors in the Custom Design platform.
- Gather more data with an improved coverage to explore the white X space, e.g., using a space filling design or Bayesian optimization,

Exercises

Data set: **DOE and Prod spectra.jmp**

1. Instead of modeling PLS scores, use Functional Data Explorer to create functional principal components (FPCs):

   a) Any data preprocessing recommended?
   b) Apply wavelet modeling.
   c) Save the first three FPCs to the data table.
   d) Compare models using PLS scores and FPCs.

2. Instead of using a boosted neural network model, try other machine learning options, such as:

   a) Support vector machines,
   b) K nearest neighbors.
   c) Generalized regression, e.g., with self-validating ensemble models (SVEM) estimation.
   d) XGBoost. This extension is available as a free JMP add-in from marketplace.jmp.com.

3. Discuss next steps for further study, including strategies for enhanced data collection, modeling, and cross validation.