

Predictor Explainer

AutoML and ExplainableAI for JMP (+Python)

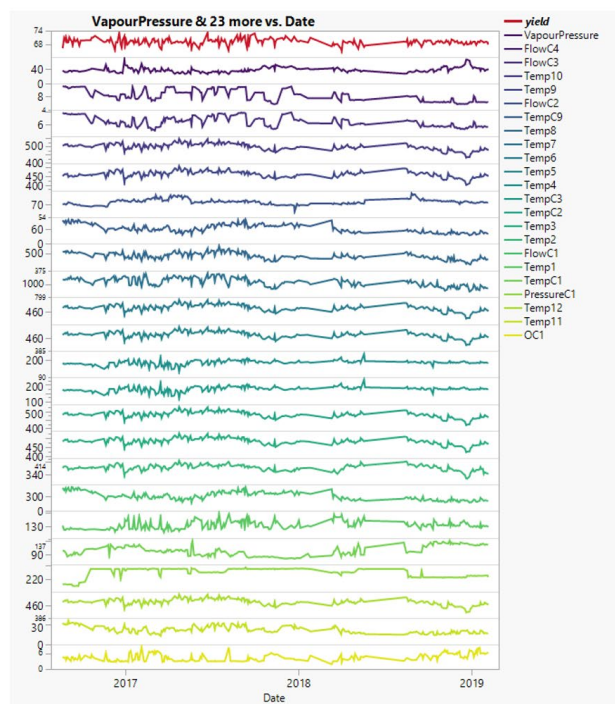
Authors: Francisco Navarro, Carlos Perez-Galvan, Juline Gilliard

License: BSD-Clause 3

URL: github.com/industrial-data/predictor-explainer

Version: 2022-09-12 (JMP 16)

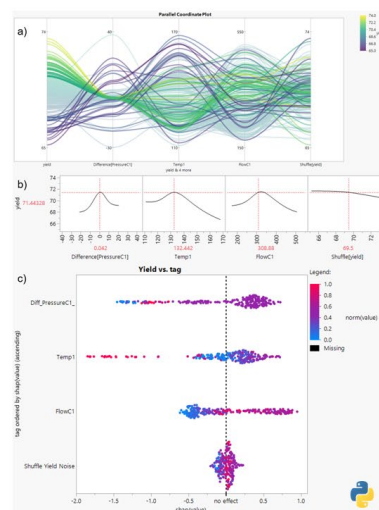
Predictor explainer automates the screening of process variables using feature engineering and machine learning (known as AutoML). Parallel coordinate plots and trends will be automatically shown to interpret the results. If PyJMP is installed (optional), SHAP plots and UMAP will be automatically calculated as well.



Predictor
Explainer



Automated screening
and selection of tags
(sensors) using
interpretable ML.



Reaction
Chemistry &
Engineering



REVIEW

Check for updates
Cite this: *React. Chem. Eng.*, 2022, 7,
1471

Industrial data science – a review of machine
learning applications for chemical and process
industries†

Max Mowbray, Mattia Vallerio, Carlos Perez-Galvan, Dongda Zhang,
Antonio Del Rio Chamosa and Francisco J. Navarro-Brul

Ref:

For further details and applications of ML applied to industrial processes, you can have a look at our open-access review: <https://pubs.rsc.org/en/content/articlelanding/2022/re/d1re00541c>

Installation steps	2
1 – Download and installation	2
2 – Distillation column (continuous process)	2
3 – Batch data analysis	6
Annex A - Creating your own version of the add-in	10

Installation steps

1 – Download and installation

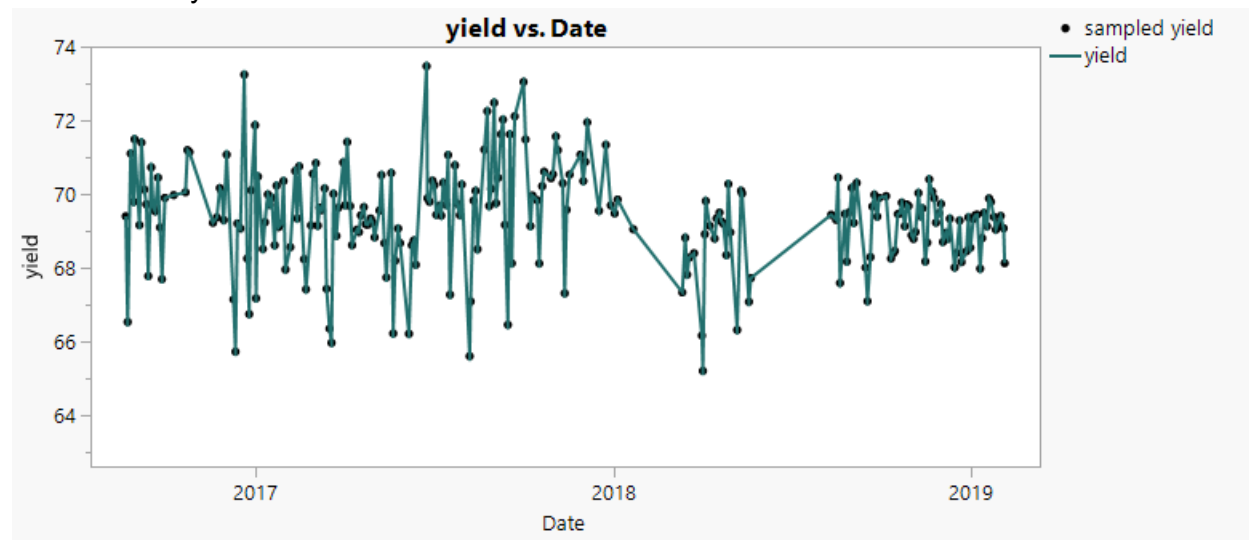
Download and click the **.jmpaddin** file to install the latest version of Predictor Explainer from:
<https://github.com/industrial-data/predictor-explainer>

If you have a previous version installed, it will be automatically removed.

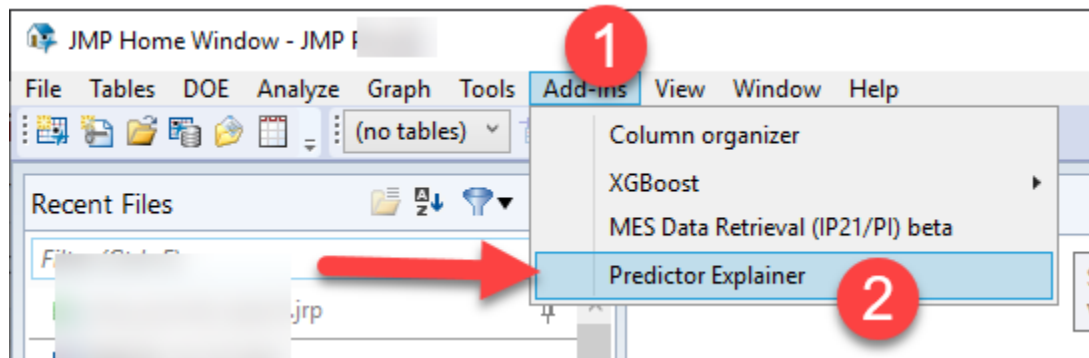
PyJMP (Python for JMP) can be installed optionally.
<https://github.com/industrial-data/predictor-explainer/pyjmp>

2 – Distillation column (continuous process)

In this example will be illustrate how to screen tags (sensor data) to quickly identify correlated factors to the yield in a distillation column.

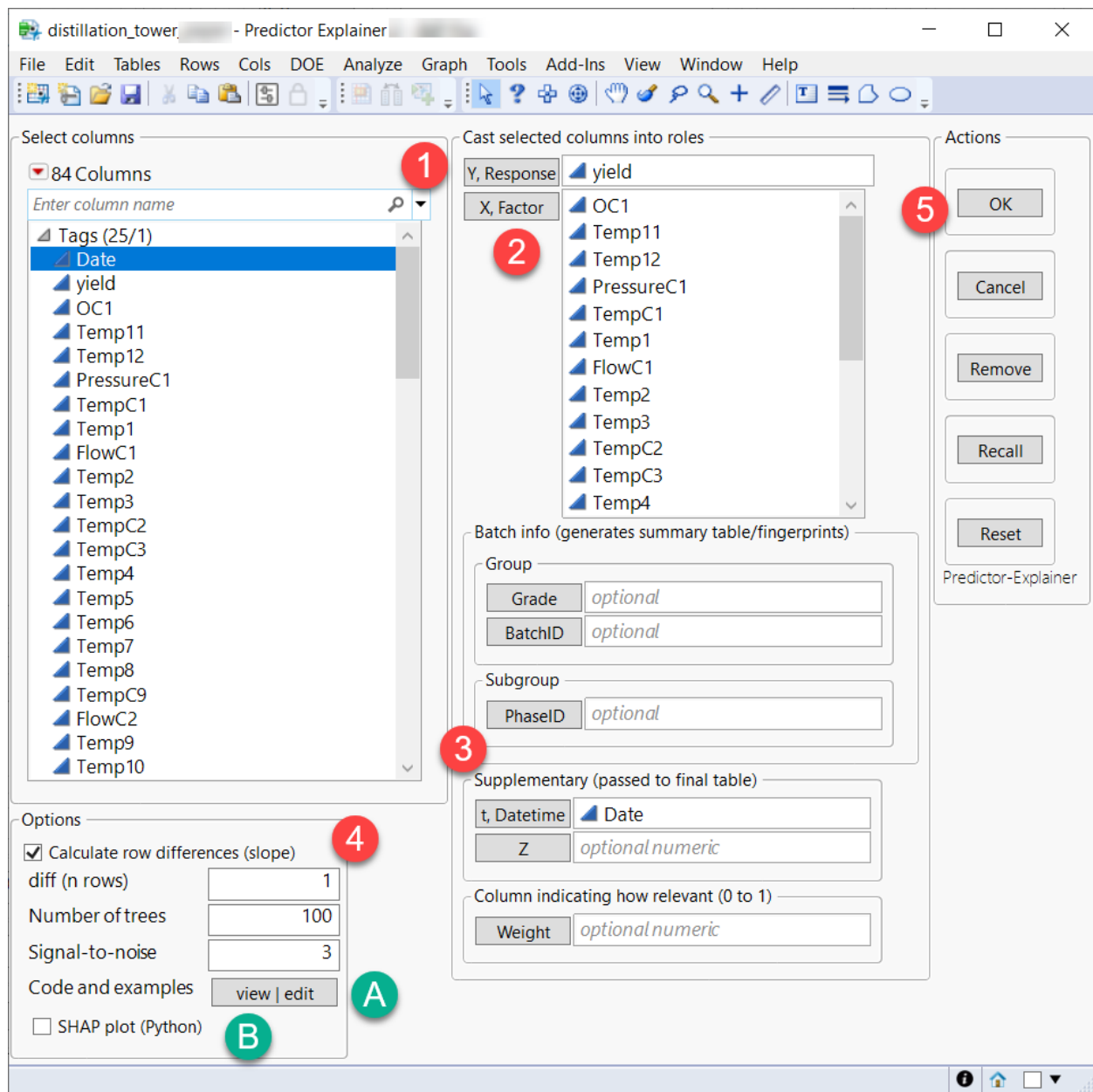


Go to the Add-ins menu and open Predictor Explorer.



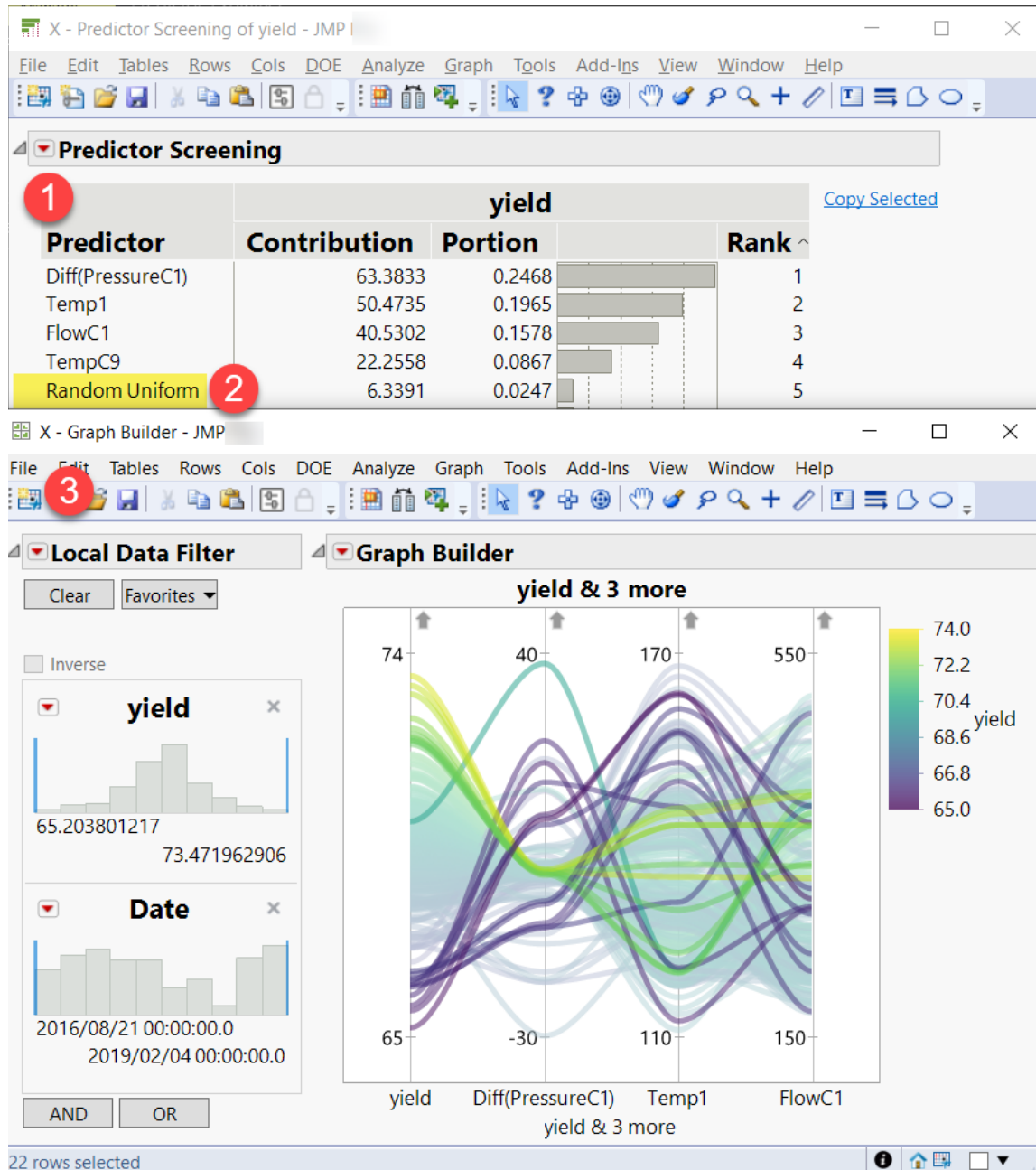
Predictor explainer contains example files. To open the folder, click in the button on the bottom left corner (see A in image below). The following steps are using the file *'distillation_column_na.jmp'*.

- (1) Select a target (Y)
 - (2) Add all the input variables or predictors (X)
 - (3) You can pass extra information such as date, supplementary information or weights. Grouping variables, will be used to create summary statistics before the analysis is done (automatic feature engineering).
 - (4) You can specify several options, such as the creating of differences for all the input variables, the number of trees in the bootstrap forest, and the threshold for the signal to noise ratio.
 - (5) Predictor Explorer will create a subset of the table and show the analysis results
- (A) The add-in folder contains other example files and a python code
- (B) If pyJMP is installed, additional options for SHAP and UMAP will be shown.

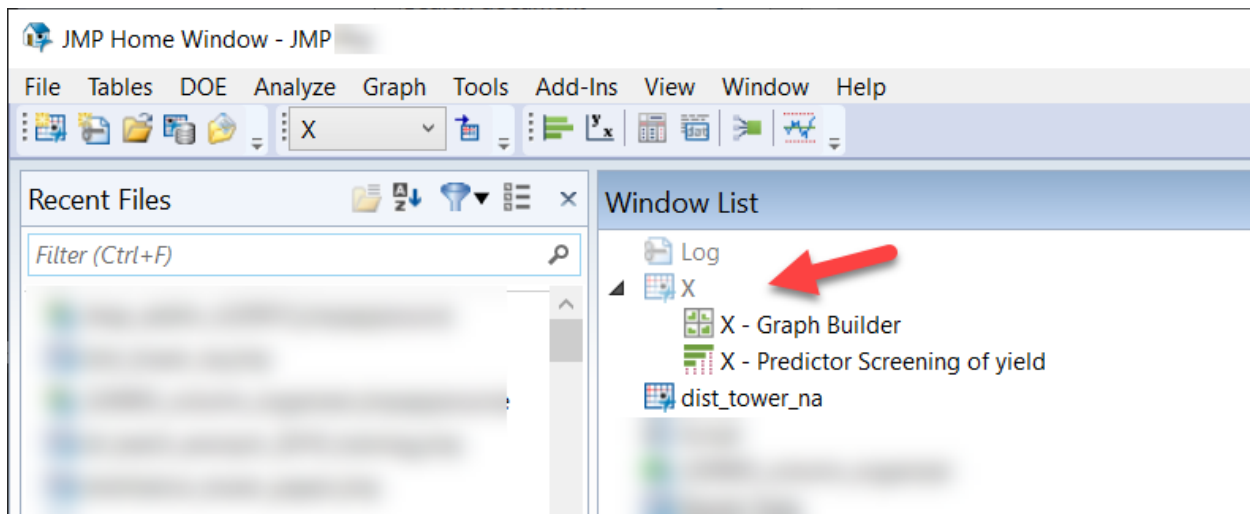


The main results are:

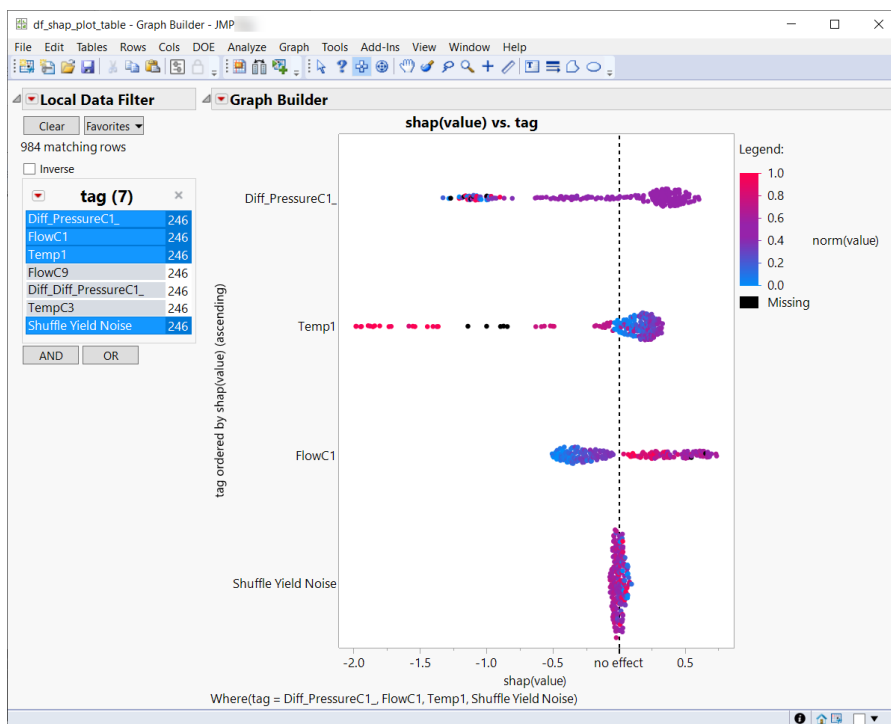
- (1) Predictor screening (bootstrap forest) of the variables introduced or pre-calculated ones using grouping or differences.
- (2) Uniform and random noise columns will be added and used as threshold to select the strongest predictors. The signal-to-noise option parameter determines how much difference in contribution will be taken
- (3) A parallel coordinate plot will be shown with the top #3 variables. If signal is not strong enough compared to noise, this plot will not be shown.



A hidden and temporal table with all the pre-selected predictors, target and time variables can be accessed via JMP home. The original analysis table won't be modified.



If PyJMP is installed and the option to show SHAP plots is activated, an interactive violin plot will appear after the analysis.



Additional hidden tables containing SHAP, UMAP and clustering results will be accessible via the home menu.

3 – Batch data analysis

Predictor Explainer can also be used to screen sensors measuring batch processes. The file named *'Fermentation_Batch_Data.jmp'* illustrates the challenge of combining unique values coming from a lab analysis with process data.



Predictor explainer will first create a table with summary statistics (also called fingerprints or landmarks).

Fermentation Batch Process - Predictor Explainer - JMP Pro

File Edit Tables Rows Cols DOE Analyze Graph Tools Add-Ins View Window Help

Select columns

13 Columns

Enter column name

- Datetime
- Yield
- BatchID**
- Batch time
- Ethanol
- Temp
- Molasses Feed
- NH3 Feed
- Air
- Tank Level
- pH
- First 4 Hours
- Validation

Options

☒ Calculate row differences (slope)
 diff (n rows)
 Number of trees
 Signal-to-noise
 Code and examples
☐ SHAP plot (Python)

Cast selected columns into roles

Y, Response
 X, Factor

optional numeric

Batch info (generates summary table/fingerprints)

Group
 optional

Subgroup
 optional

Supplementary (passed to final table)

optional numeric

Column indicating how relevant (0 to 1)

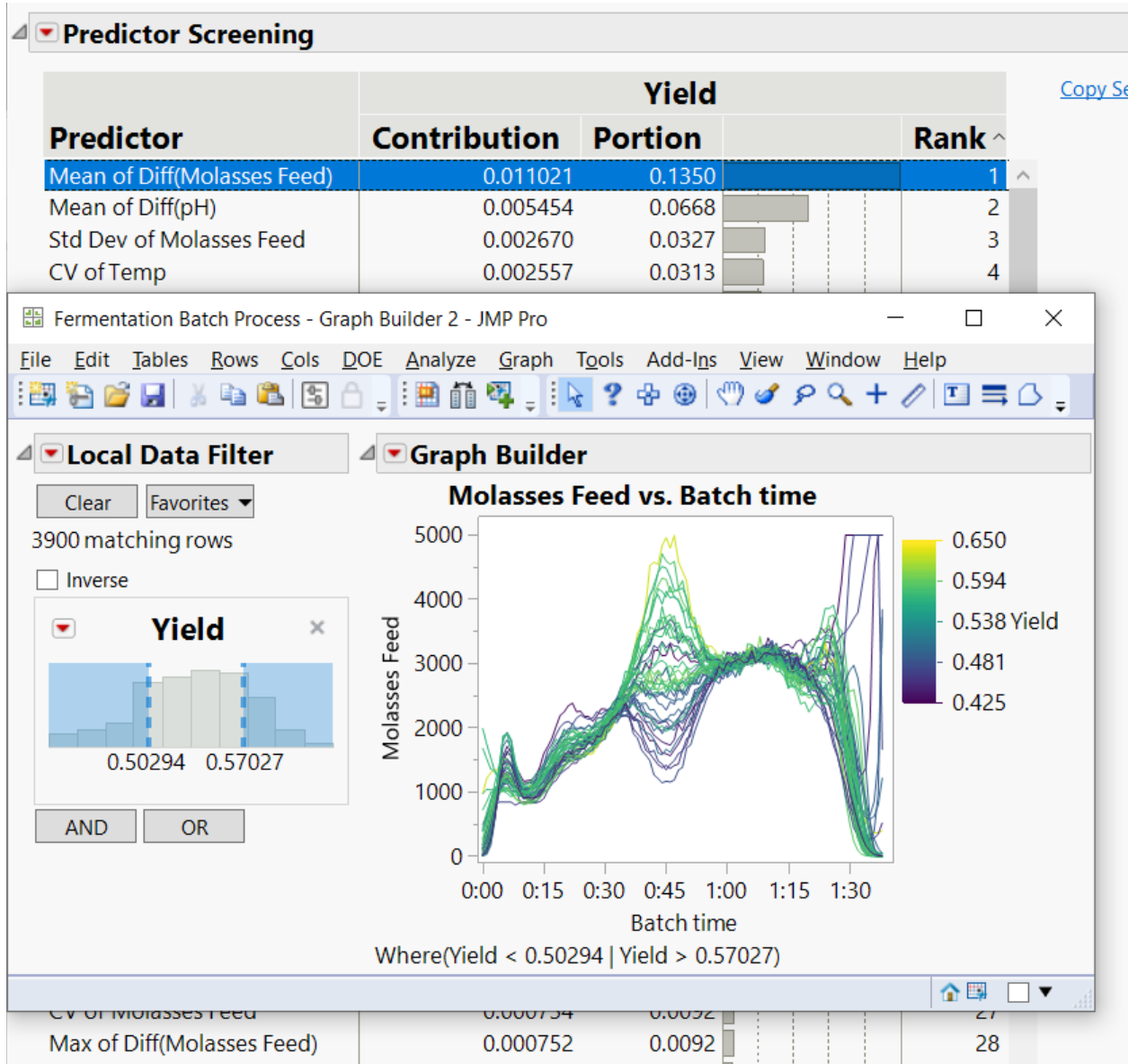
optional numeric

Actions

Predictor-Explainer

If there is information about product (grade) or phase (stage) of the batch, these will be also used to generate more granular summary statistics. Using the noise contribution as a filter will eliminate all calculated features below it. When PhaseID is introduced as numeric and ordinal column (1, 2, 3...), an automatic aligning of the batch will be performed and shown.

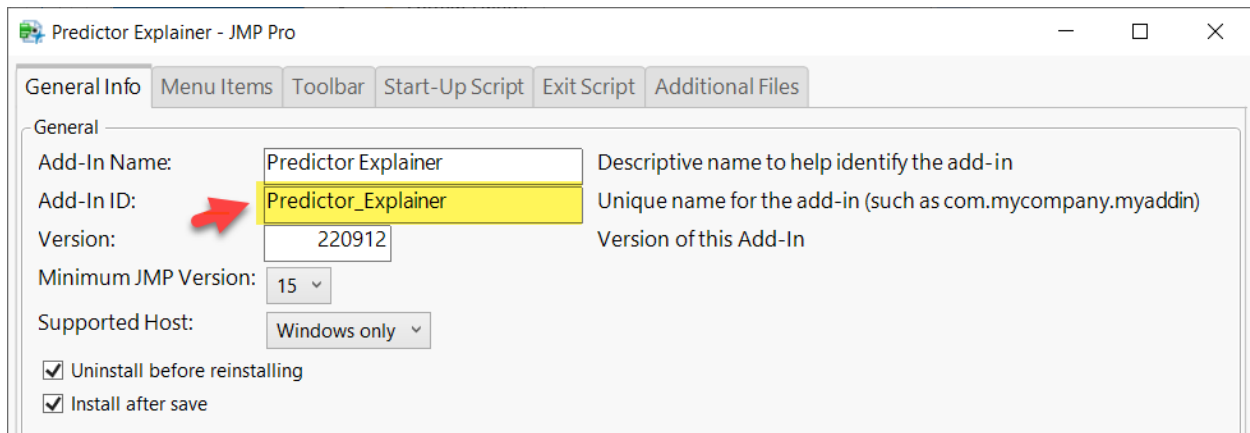
In the example, Predictor Explainer identified the strongest sensor (tag).



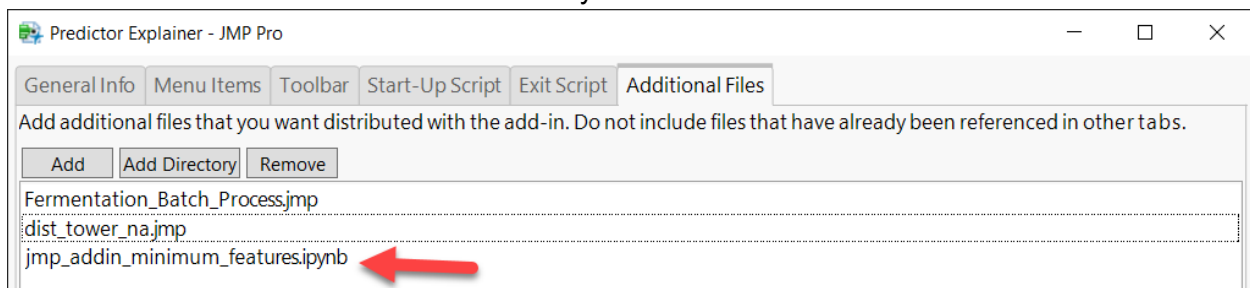
Annex A - Creating your own version of the add-in

If you want to distribute or modify a new version of the JMP addin, there are two important things to consider.

1. Changing the Jupyter notebook will not require changing the JMP add-in itself, as JMP calls the Python code independently via PyJMP. JMP will generate temporary files to communicate with Python and then read the results. The Jupyter Notebooks contains all the steps, to access it open the folder clicking the button on bottom left corner. There is one executable called open_notebook.bat which will allow you to modify the jupyter that is executed.
2. When modifying the JML add-in source code and saving it (exporting application), make sure to keep the same Unique ID:



As well as add the Notebook and other files you want to distribute with the add-in.



In case you have suggestions, errors, or success stories (!) contact us

<https://github.com/Industrial-data/predictor-explainer>