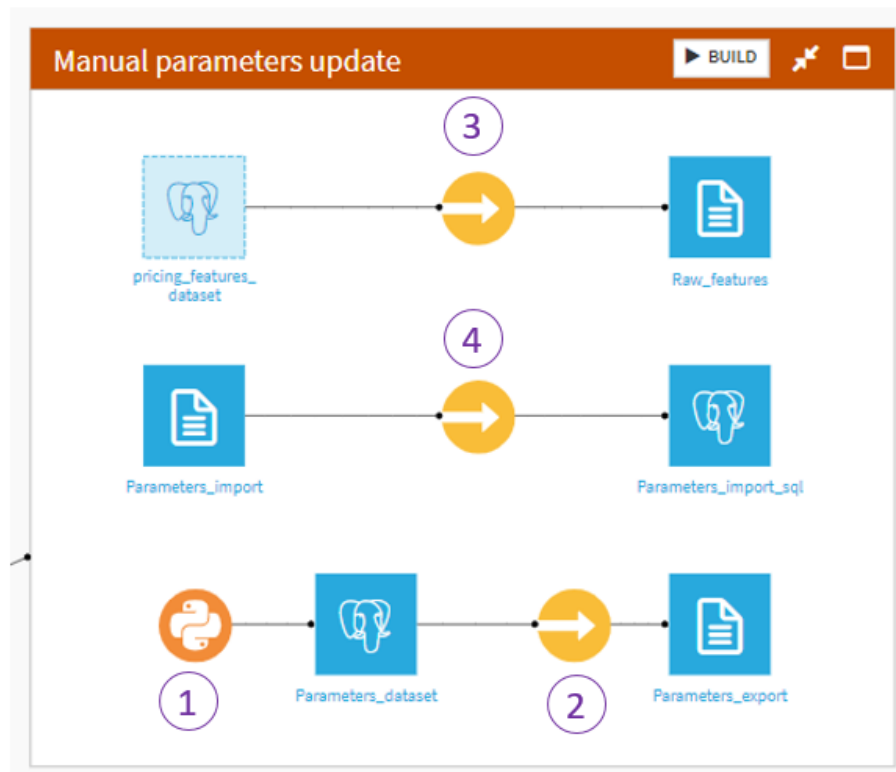


9. Variables (parameters) update

Some of the [variables](#) of the project can be updated in an autonomous way by the users. This allows them to change some of the business rules without any action required from a developer or data scientist.

Elements of the Dataiku project used for this process are stored in the "Manual parameters update" Flow zone : [SpP zone](#), [NC zone](#).



The variables are sent in a dedicated GSheet (SpP file, NC file) accessible by the users through the following steps :

Export of current variables from the Dataiku project to GSheet

The first recipe (1 in the image above) uses the dedicated defined variables (refer to the [External variables update](#) section of the variable documentation) and functions in the *utils* module of the code base to store the parameters in a dataset.

This dataset is then synchronized on a hidden *From_Dataiku* tab of the Gsheet (2).

At the same time, we also export the most recent state of the *pricing_features_dataset* on the *Dataiku_raw_input* tab of the GSheet (3). This is useful for the users to be able to **see what data is behind the technical parameter names** that are used in the other tabs. It can also be used to do some testing on the data that is consumed by the models, since *pricing_features_dataset* contains the features used after all the pre-processing steps.

This export step is conducted at the **end of any automated run** using the scenario "8 - External parameters export" to keep the GSheet updated.

Update of variables in Gsheet

In the GSheet, the two main tabs for the users are the following :

- "*Synthesis and global parameters*" : contains all the global variables that can be updated for all families at the same time. It also contains their description and original value.
- "*Family parameters*" : contains the variables to update on a family basis

The "*Dataiku_raw_input*" tab can also be used to view the latest data used in the Dataiku project.

Hidden tabs :

- "*From_Dataiku*" : Data coming directly from the Dataiku project containing the current values and description of the variables.
- "*To_Dataiku*" : Data being sent back to Dataiku after the users update, following the same format as the "*From_Dataiku*" sheet.
- "*Hard_boundaries_list*" : Contains the list of hard boundaries by family, extracted from one of the rows of the "*From_Dataiku*" table.

- "*Recommendation_cap_list*": Contains the recommendation caps by family, extracted from one of the rows of the "*From_Dataiku*" table.
- "*Volume_threshold_list*": Contains the volume thresholds by family, extracted from one of the rows of the "*From_Dataiku*" table.
- "*Families_list*": Contains the list of validated families, extracted from one of the rows of the "*From_Dataiku*" table.

File step-by-step process for maintenance

This file contains **several formulas that could require maintenance**. By default, most of the cells of the user sheets are **locked** in order to not impact the formulas used.

Here are the detailed processing steps of the file :

Single values

- In "*Synthesis and global parameters*", input a variable name in column A (hidden). The name should match a row from the "*variable*" column of the "*From_Dataiku*" sheet.
- The *description* and *value* cells are then updated using VLOOKUP formulas based on the variable name in column A.
- The *new value* is directly read from the "*To_Dataiku*" sheet to be sent back to Dataiku.

Validated families list

- The list is split and transposed using an ARRAYFORMULA formula in the dedicated hidden sheet "*Families_list*".
- The validated families are then retrieved in column A of the "*Family parameters*" sheet where the users can select which ones to include in the next run (column B).
- A TEXTJOIN formula is then used in "*Synthesis and global parameters*" sheet to list the families that have been ticked to be included in the next run.
- Finally, the generated list is read from the "*To_Dataiku*" sheet to be sent back to Dataiku.

Hard boundaries dictionary

Note : hard-boundaries are currently not part of the variables the users can adapt themselves. It indeed can have some consequent impacts on the results that they can not monitor properly with the current tools they have access to (average number of comparables by product family for example). They are still displayed as an additional information and can be adapted through technical intervention and validation of the results.

- The dictionary is split and transposed using nested ARRAYFORMULA formulas in the dedicated hidden sheet "*Hard_boundaries_list*".
- The hard boundaries are then retrieved in column G and following of the "*Family parameters*" sheet.

Recommendation cap dictionary

- The input dictionary is split and transposed using nested ARRAYFORMULA formulas in the dedicated hidden sheet "*Recommendation_cap_list*".
- The original values are then retrieved in column C of the "*Family parameters*" sheet.
- Users can input values in column B, which are then read in the "*Recommendation_cap_list*" sheet in columns F to H. This is where we create the output dictionary in a scalable way, meaning we can add or remove families without disruption of the format.

Volume threshold dictionary

- The input dictionary is split and transposed using ARRAYFORMULA formula in the dedicated hidden sheet "*Volume_threshold_list*".
- The original values are then retrieved in columns E and G of the "*Family parameters*" sheet.
- Users can input values in columns D and F, which are then read in the "*Volume_threshold_list*" sheet in columns I to K. This is where we create the output dictionary in a scalable way, meaning we can add or remove families without disruption of the format.

Import of GSheet updated variables in Dataiku

To import back the updated variables in Dataiku, we start by synchronizing the "*To_Dataiku*" tab of the GSheet to a dedicated "*Parameters_import_sql*" dataset (4).

Then, in a dedicated scenario "*0 - External parameters import*" ran at the **beginning of any automated run**, we run a custom Python step and use a function from the code-base to read the "*Parameters_import_sql*" dataset and set the Dataiku variables.



Beware that in the *To_Dataiku* sheet, the **last record** should **not have** a *technical_path* (column B) being a **sub-dictionary** of the variables (e.g. *model.local_price_recommendation* where *model* is a sub-level of the variables dictionary). This is a limitation of the code used to read the output and update the variables.