1. Data preparation

- CPC definition
- Target
- Price drivers
- Forecasts and historical data
- [Novecare] Detailed processing steps
- [SpP] Detailed processing steps
- Manual input
    - Update the inputs from Gsheet directly (Novecare only)

CPC definition

You will find the term "**CPC**" a lot in the following documentation as it is the **main level of granularity** used through the application. This means that most of our datasets will have one record by CPC of a given GBU.

The Customer Product Combination (CPC) is the identifier representing **a specific product sold to a specific customer**.

The product and customer definition here varies from one GBU to another.

For example :

- For **Novecare**, the product considered is the *material* and the customer is the *shipto*.
- For **SpP**, the product considered is the *material_group* and the customer is the *soldto*.

Target

The target represents the data we are trying to **optimize**, in this case the **unit price** of a CPC (Customer Product Combination).

The unit price we use in our models is the result of several computation made in our data preparation steps, we could summarize it as follows :

For **Novecare**:

- Gathering and **aggregation** of the CPC forecasted **sales** and **volumes** on the **next 12 months.**
    - Computation of the resulting **forecasted unit price**
- Gathering of the CPC **last invoiced price** from the forecast data.
- Gathering and **aggregation** of the CPC historical **sales** and **volumes** on the **last 12 months**.
    - Computation of the resulting **historical unit price**
- Gathering of the CPC **last invoiced price** from the historical data.
- Select the first non-zero unit price following this order :
    - Forecasted unit price
    - Last invoice price from forecasts
    - Last invoice price from historical data
    - Historical unit price

For **SpP**:

The rule used to define the unit price is as follows:

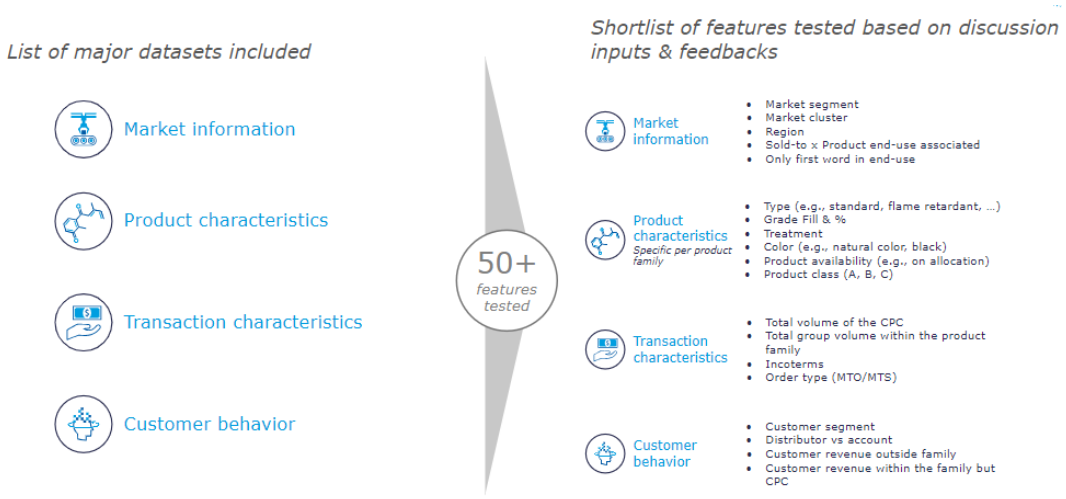
- Average of **volume-weighted prices** for the **last 3 months with sales**.

Note : as of now, this unit price includes all costs : fixed and variable.

More details are available below on how these computations are included in our global data preparation flow.

Price drivers

To select the final list of the most relevant price drivers, we collected, built and tested more than 50 features:



These price drivers are coming from several data sources described below.

Forecasts and historical data

The main data source we are currently using is the Pricing Data Lake in Big Query, especially the two following datasets :

- **V_FACT_sales_forecast_enriched_current** : Forecasts data.
- **V_FACT_sales_history_cpc_last12months** : Historical data for the past 12 months.

These datasets include :

- Sales and volume **measures** that are also used to generate the **unit price** used as a target (see dedicated § above) for our models.
- **Dimensions** used as features / **price drivers** (see dedicated § above).

[Novecare] Detailed processing steps

[SpP] Detailed processing steps

Manual input

All manual data sources are gathered in a spreadsheet specified by GBU and under the responsibility of the business.

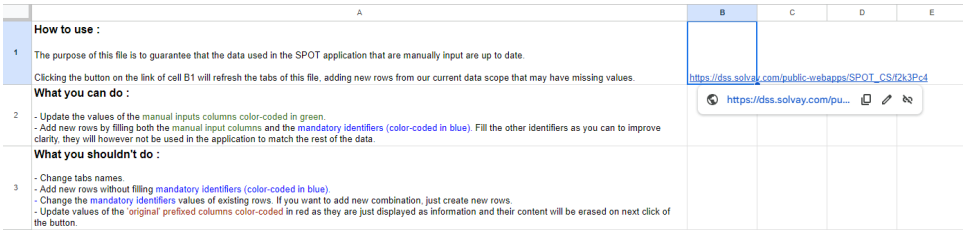
For **Novecare**, this following GSheet is used to process the manual inputs for **product groupings**, **manual regions** and **manufacturing plant groups**.

For **SpP**, this following GSheet is used to process the manual inputs for **product taxonomy** and this one for **manual regions**.

Update the inputs from Gsheet directly (**Novecare only**)

Here are the steps to follow to use this manual process :

1. The user clicks on the button of the link displayed in the "README" tab to **trigger an update of the content** of the file (this will take a few seconds and is finished as soon as the button re-appears).



	A	B	C	D	E
1	**How to use :** The purpose of this file is to guarantee that the data used in the SPOT application that are manually input are up to date. Clicking the button on the link of cell B1 will refresh the tabs of this file, adding new rows from our current data scope that may have missing values.	https://dss.solvay.com/public-webapps/SPOT_CS/f2k3Pc4			
2	**What you can do :** - Update the values of the manual inputs columns color-coded in green. - Add new rows by filling both the manual input columns and the mandatory identifiers (color-coded in blue). Fill the other identifiers as you can to improve clarity, they will however not be used in the application to match the rest of the data.				
3	**What you shouldn't do :** - Change tabs names. - Add new rows without filling mandatory identifiers (color-coded in blue). - Change the mandatory identifiers values of existing rows. If you want to add new combination, just create new rows. - Update values of the 'original' prefixed columns color-coded in red as they are just displayed as information and their content will be erased on next click of the button.				

2. In Dataiku, it will launch a scenario that will retrieve the data of the latest run and send them in the several tabs of the Gsheet.

   Let's take the example of the "Regions" tab :

1	country_shipto	product_family_h4	original_manual_region	manual_region
462	Türkiye (TR)	Alkoxylates	Rest of the world	Rest of the world
463	Ukraine (UA)	Alkoxylates	Rest of the world	Rest of the world
464	Uruguay (UY)	Alkoxylates	Rest of the world	Rest of the world
465	Uzbekistan (UZ)	Alkoxylates	Rest of the world	Rest of the world
466	Venezuela (VE)	Alkoxylates	Rest of the world	Rest of the world
467	Oman (OM)	Alkoxylates	Rest of the world	Rest of the world
468	Curaçao (CW)	Alkoxylates	Rest of the world	Rest of the world
469	Uganda (UG)	Alkoxylates	Rest of the world	Rest of the world
470	China (CN)	Esters		
471	India (IN)	Sulfates & Sulfonates		
472	Canada (CA)	Formulations & Blends		
473	Usa (US)	Formulations & Blends		

   We can see that new records appeared with countries that do not have a *manual_region* value for the given *product_family_h4.*
3. The user can now add the right *manual_region* in the green background column or update an existing value. *Note : The original_ prefixed columns will always display the current value to help the users potentially reverting changes.*