# 4. Data encoding



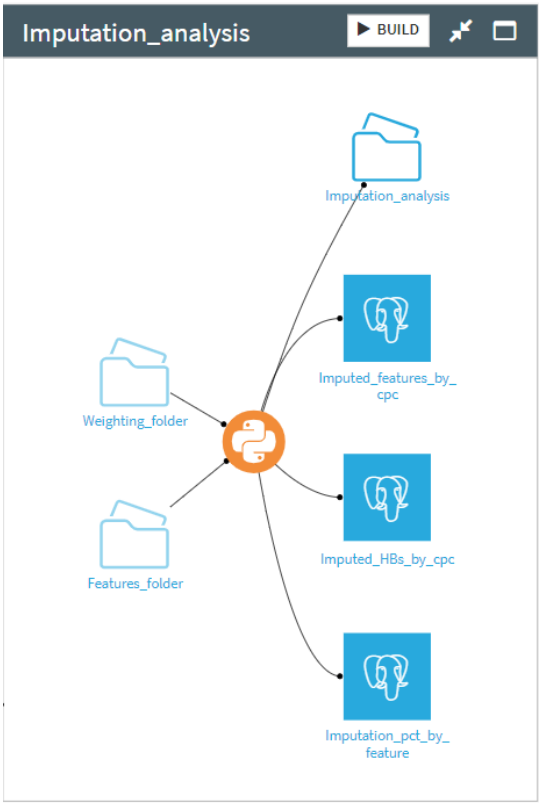
## Imputation

Before moving on to the data encoding stage, we first impute the missing values in the features using a dictionary variable in Dataiku variables, which is used by the "*simple_imput()"* function in the Python encoding recipe.

The details of the imputation method by GBU and by feature are described here:

- **Novecare** imputation details
- **SpP** imputation details

To monitor the percentage of imputations per feature and per CPC, we added a specific zone in the flow named "**Imputation_analysis**":




In this zone we created three separate datasets, as follow:

- Imputed_features_by_cpc: Contains imputed values (based on the adopted method defined above) for each CPC for the included features in the corresponding family. Two metrics are controlled in this dataset, the first being "***tot_features_weights***" which computes the impact of the imputed features for each CPC based on the average weights (from the first model) of these features. The second is "***pct_imputed_cols***", which calculates the percentage of imputed features in relation to the features included in the corresponding CPC family.

==> The checks carried out in this dataset compare these metrics with a fixed threshold of 0.4 for the first metric and 0.5 for the second (these values can be modified in the check parameters).

- Imputed_HBs_by_cpc: Contains imputation statistics only for the features included as hard boundaries (number and percentage of imputed hard boundaries for each family by CPC).

==> The check sends a warning or an error (depending on the type of message defined in the dataiku variables) if the "pct_imputed_cols" is greater than a threshold (set to 0.5 in the check parameters).

- Imputation_pct_by_feature: Calculates the percentage of imputation for each features (including HBs) by family.

==> The check verifies whether the imputation pct for a given feature is greater than the defined threshold (set at 0.5 in the check parameters).

# Collapse categorical

In the Python encoding recipe we also collapse categorical features using the "*collapse_categorical()"* function, in which we define a threshold to collapse modalities that have fewer values than this threshold (*the default value is 5*).

# Encoding

***Vocabulary :*** *target CPC vs. target of the model*

*In order to understand this section, it is important to clarify two different usages of the word "**target**". Indeed, since the machine learning part of the project includes two successive models, the usage of **target** is different for the two of them.*

- *Our first machine learning model is outputing the importance of each of our **features** on the **price** of the CPC (customer/product combinations).*
  - *All the features are also called **variables**. They include for example the region of the customer, its segment, market cluster or characteristics on the products.*
  - *The price is also called **target** as it is what we are trying to explain here.*

- *Our second machine learning model is outputing similarity distances between CPCs (using results of the first model).*
  *This then allows us to rank every CPC (comparables cpc) in comparison to a specific one : the **target cpc**. The lower the distance is, the closer the comparable is to the target cpc.*

---

For our models to work and compute the distance, we need to have numeric values only as input. Since we also have categorical (e.g. region and incoterms features) features in our original data, we have to transform them to numerical data.

This is a common step in machine learning and is called **"encoding"**. In our case we are using a specific encoder for this which is named **"target encoder"**.

Here is how it works :

- Each categorical feature modality is replaced by a numeric value.
- The encoded value here represents the average price for every CPC having the initial value. The price considered here has a "log10" function applied. It is called **target encoder** because our target is the price (with log).
  This allows us to replace the initial region by a feature containing both informations on regions and prices.

**Example** on the "Incoterms" feature for *Sulfosuccinate_Sulfosuccinamate* family :

| Initial value | Encoded value |
|---|---|
| PPD | 0.72 |
| DDP | 0.41 |
| COL | 0.80 |
| PPD | 0.72 |
| CIF | 0.64 |

**In the example above :**

- Any CPC with "PPD" as region will have a value of 0.72.
- The average price for COL is greater compared to other incoterm values