AI smart price engine walk-through

The objective of this page is provide simple explanations for the business to understand all the steps of the modelization using few technical items.

1.Data 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 product taxonomy 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 "Region" feature for Amodel family :

Initial value	Encoded value
Americas	0.99
GCN	0.95
EMEA	0.91
Other APAC	0.91

In the example above :

- Any CPC with "EMEA" as region will have a value of 0.91.
- The average price in Americas is greater compared to other regions
- Note that CPCs with "other APAC" will have the same value as "EMEA". This allows us to consider them totally comparable because their average price is equivalent.

From a categorical feature with no information about order and proximity between modalities, we obtain an ordered numeric variable usable for machine learning model and the similarity distance calculation.

2. First model : Compute feature importance on price

Note : one model is built for each of the product family, every family is totally independant.

The model used here is a regression model (LightGBM specifically).

The **usual** objective of a regression model is to predict the value of the numeric **target** (price with "log" transformation) using the input numeric features (result of previously explained encoding step).
To do this, some CPC are used to train the model, and others to test the performance. An optimization (also called fine-tuning) is done to find the best parameters of the model for each family.

The R^2 metrics is used to measure the **model performance**, result is generally between 0 (bad) and 1 (perfect). We consider the model good enough between 0.4 and 0.9.

Note : the objective is not to have a perfect model, because fitting too well the training data often leads to a bad generalization.
In simpler words, it means that the provided data are too specific and detailed and while it allows the model to perform better on the training set, it will fail its prediction with any slight change in the new data we will provide at each campaign.

If R2 are too low, this could mean two things :

- We are missing some pricing levers (features) to explain price successfully. The ones we provide are not enough to have a good prediction on price.
- The dispersion of prices cannot be explained by data (due to human behavior, specific negotiation with the customers, etc.)
  For example, we can end up with 2 CPCs with exactly the same values for all features, with different prices.

In this use-case, we do not use the output of the model to predict prices directly.

What we are interested in is to understand the **importance** of each of our pricing lever (feature) in predicting the price.
This is done by computing SHAP values when running the model. That is also why we can accept average R2 score.
This feature importance will then be used by the second model as coefficient to find neighbors for each CPC.

Next section describes the model outputs that should be reviewed to validate the modelization step.

3.Modelization outputs

3.1 R^2

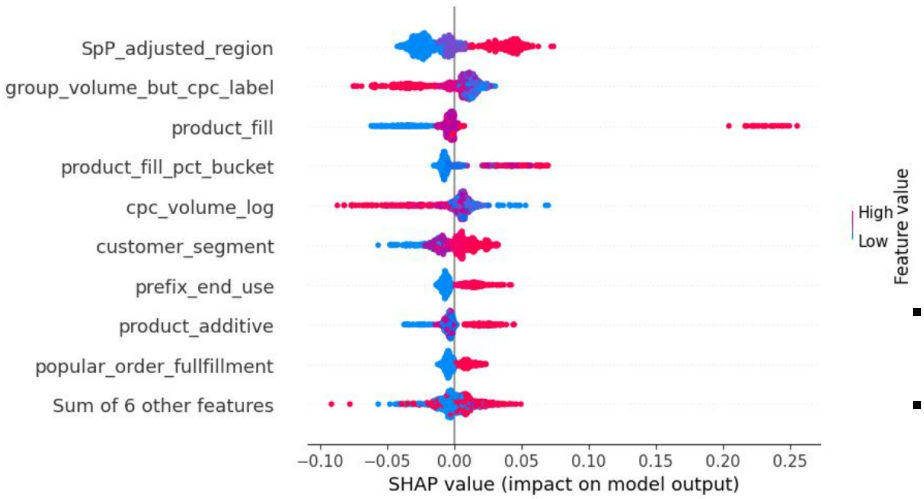
Measures the prediction performance of the model. We can compare it with the previous campaign and see if they are stable.

Any big decrease could represent a data issue. If data seems clean and a decrease still appears, models have to be retrain (with a grid search) to find optimal parameters.
If no decrease, retraining and optmizing parameters should be done once a year.

```
family,r2
ryton,0.505
veradel,0.815
amodel,0.588
solef suspension,0.25
ketaspire,0.583
torlon,0.65
kalix,0.866
ixef,0.695
udel,0.477
radel,0.698
tecnoflon ffkm,0.548
tecnoflon excluding ffkm,0.91
fluids_neutral & fluids_functional & solvera,0.869
halar,0.774
hyflon,0.426
pvdc,0.722
```

3.2 Features importance : Shap values

Amodel example :



Shap values, sorted by importance

- Each dot represents a CPC
- Color is red when a CPC has a high value for the feature (y-axis), and blue for a low value.

These are numeric values coming from the encoding step.

- The SHAP values (x-axis) represent the impact of a feature on the price prediction. This can be understood as the deviation from the average prices of the family due to the feature.
    - Strength of the SHAP value is that it is able to isolate the impact of a specific feature, ignoring all the other pricing features input in the model.
    - Since we are modelizing the price with a **log10** applied, and not the raw value, we can not direcly interpret the x-axis as a "% price variation" on this graph.
      We need to apply the log10 inverse function (which is 10^x) to bring it back to the "normal" scale.
      We can then read it as a real percentage of price deviation from the average of the family.

      Example on the "SpP_adjusted_region" feature (please refer to the table in Data encoding part above) :
        - When a CPC has a low feature value for the region (APAC and EMEA), the dot is displayed in blue. From the x-axis, we read a SHAP value at nearly -0.03. After applying the transformation (10^-0.03), we retrieve an average impact of -7% on price.
          This means that being in the APAC and EMEA regions brings the price down to 7% compared to the other CPCs of the families.
        - When a CPC has a high value (Americas) the price is around 0.05 higher (12% after transformation).
    - For the volume, we can see that high volumes (in red) have negative impact on price, which is expected.

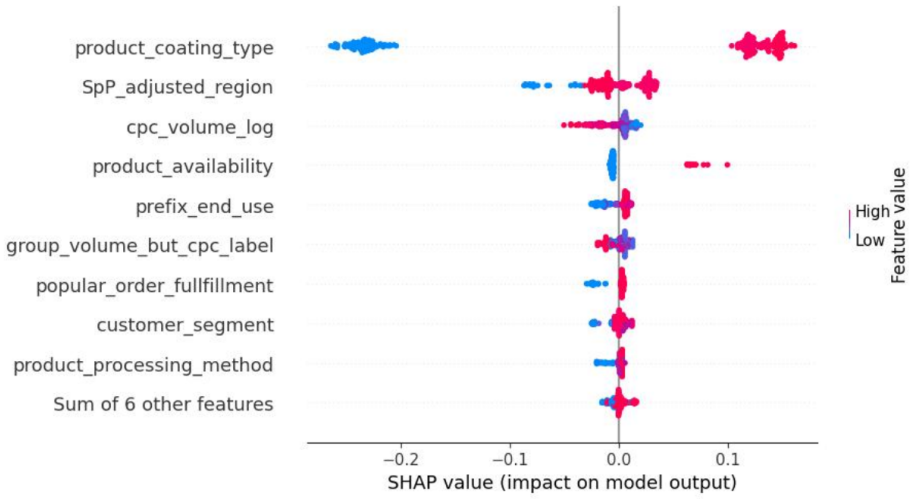
SHAP values are then used to define the weight (or importance) of every pricing lever.

feature_name	feature_weight
SpP_adjusted_region	0,18
group_volume_but_cpc_label	0,12
product_fill	0,12
product_fill_pct_bucket	0,09
cpc_volume_log	0,09
customer_segment	0,09
prefix_end_use	0,07
product_additive	0,06
popular_order_fullfillment	0,05
market_cluster	0,04
rev_outside_family_log	0,04
product_treatment	0,03
product_base_resin	0,02
product_availability	0,01
product_compounded	0,00

Final feature importance (weight).

- We first compute the absolute average of every CPC SHAP value by feature (average SHAP value of every dot in previous graph).
- We then normalize the values between features so all the weights in the table sum to 1. It allows us to compare all the different features that were independent since then.
  For example, the region is our most important feature and explains 18% of the dispersion around the family price average.

Halar example :



Let's observe the "**product_coating_type**" feature : We can see two distinct clusters.

- The blue one which represents the "not relevant" modality has an impact of -0.225 (-40% after transformation) on the price.
- The red one which represents the "Primer" / "Top coat" / "Standalone" modalities has an impact of 0.135 (33% after transformation)

The impact difference here is really high between the groups, they have therefore be considered **not comparable** one with the other.
This lead to the creation of what we call a "**hard boundary**".
When trying to find comparables in the next steps, we will **filter out** from the comparable set any CPC that has a different value than the target CPC for product_coating_type.

4. Second model : Find comparable CPCs

To find comparables, we compute a similarity distance between all CPCs two by two. This computation is based on the numeric values of the CPC features on which we apply the **feature weight from the first model** as a factor.

- The algorithm used for this is named "K-nearest neighbors".
- The computed distance is a "cosine" distance, which is commonly used in data science.
- If 2 CPCs have similar values for an **important variable**, it is more impactful than CPCs having similar values on a **low importance variable.**
- **Volume** feature is **excluded** from similarity distance calculation. Indeed, volume is treated after in its own "volume adjustment" step and the objective here is not to find comparables which have similar volume, but similar characteristics.

For each target CPC, all comparables are ranked according to the similarity distance, and we exclude comparables outside of the hard boundaries.

In the end, we keep the 10 closest comparables (the ones with the lowest similarity distances) for final price calculation and display in the WebApp.

5. Volume adjustment

Volume being **excluded** from the comparable selection, we can have big gaps between a target and its comparables. That is why we need a specific volume adjustment step.

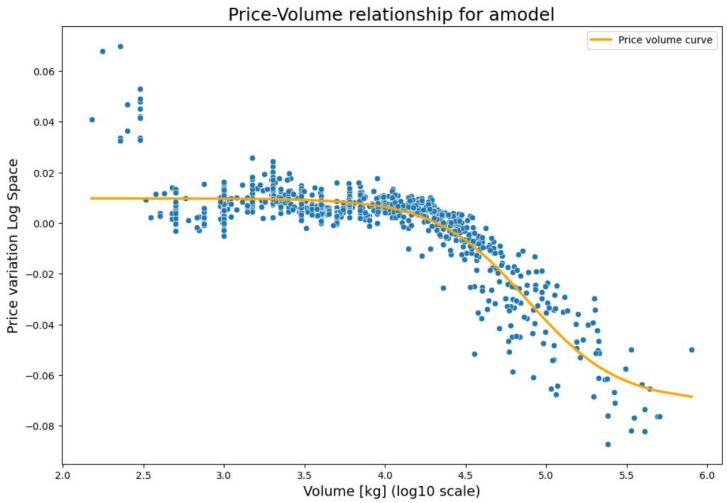
The objective is to adjust the prices of comparables to answer the following question : **"what would be the comparable cpc price if it was sold at the same volume as the target cpc ?"**

Example for Amodel :

First steps : computing the adjustment function for each of the families

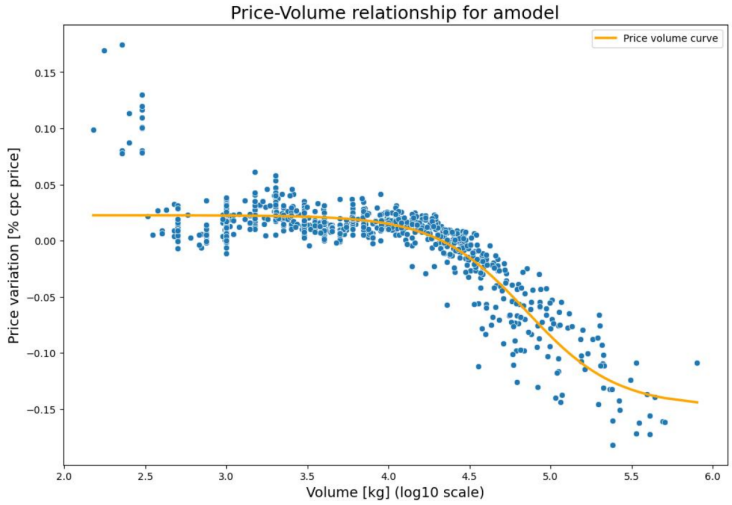
- We start from the SHAP value of the volume (please refer to section 3.2 above).
  Each dot here represents a cpc and the x-axis (SHAP value of cpc_volume_log) goes from -0.09 to +0.06

- We add the volume as a second dimension on the X axis to create a scatterplot.
  - On the Y-axis, we retrieve our Shap values with the scale between -0.09 and +0.06.
  - Each dot still represents a cpc.
  - We can see that the SHAP values decreases as volume increase.



- We find the best fit function to the scatterplot and display it as a curve (in yellow above).

- Finally, we apply the log inverse function (10^x) to the y-axis to obtain a result that we can interprete. The y-axis value is now a price variation due to volume, compared to the average prices of the cpc of the family.
  Note : the shape of the curve does not change, only the Y-axis is modified

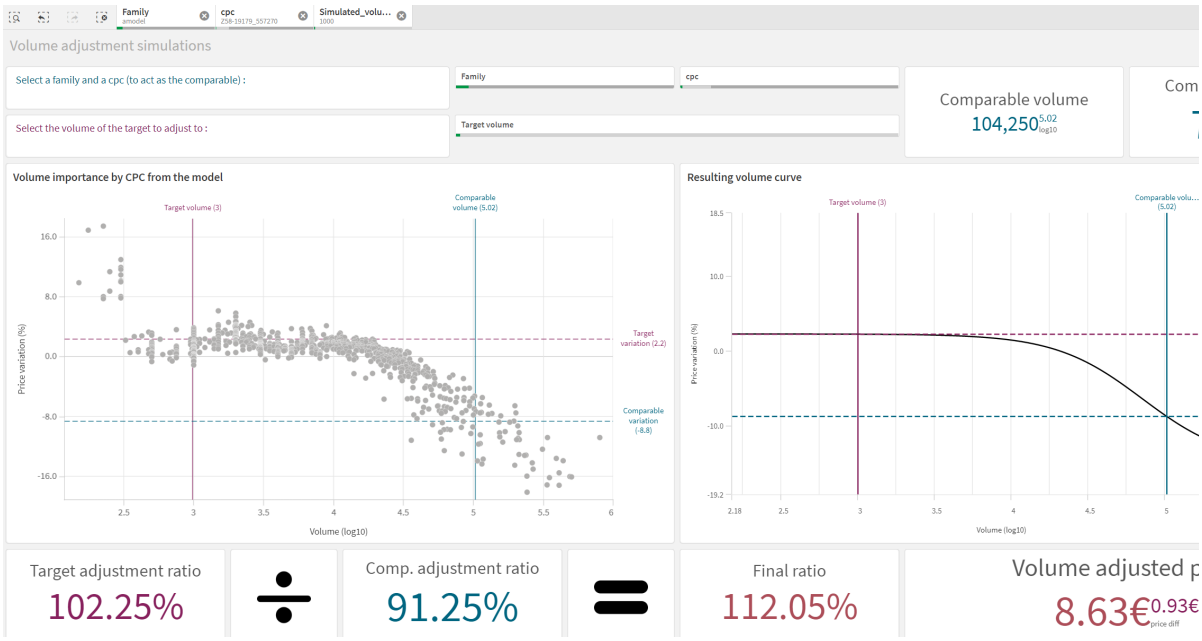


Then, adjusting every comparable using the fit function.

These steps are simulated in the "Volume adjustment simulations" tab of the dashboard for a better understanding based on real values of the current run.

To use the tab, you need to select a family and a cpc that will be used as a comparable. Then, you can enter the volume you want to adjust to. This simulates the volume of the target cpc.
By changing the volume, you can see how the volume adjusted price is affected.

Amodel CPC example Z58-19179_557270 :



- The selected cpc has a volume of 104 250 and we are comparing to a 1000 volume target.
  When adjusting, we expect to have an increase in price since the target has a lower volume.

- The scatterplot on the left is a representation of all the cpcs of the family and their **price variation due to volume** (based on SHAP values from the first model).
  It is the **same scatterplot** as the one displayed above.

- The line chart on the right is the **fit function** for the family, originated from the scatterplot.

- **X-axis** (volumes) on both of the graphs are on a **log scale** to make it more readable and less prone to outliers.

- **Y-axis** (price variations) on both of the graphs are **out of the scale** and represent understandable price variations. That is why the scale is similar to the second scatterplot above, and not the first one.

- Our **comparable** cpc :
  - Has a volume of 104 250, which equals **5.02** in log10 scale.
  - Crosses the fit function at an adjustment value of **-8.75%**.
    - This means that its volume lowers its price by 8.75% compared to the family average.
    - This leads to a ratio of **91.25%** (= 100% - 8.75%).

- Our **target** cpc (not defined since it works for any target with this volume) :
  - Has a volume of 1 000, which equals 3.0 in log10 scale.
  - Crosses the fit function at an adjustment value of **+2.25%**.
    - This means that its volume increases its price by 2.25% compared to the family average.
    - This leads to a ratio of **102.25%** (= 100% + 2.25%).

- To adjust our comparable to the target volume, we compute the **ratio** between the target value and the comparable value.
  - 102.25% / 91.25% = **112.05%**

- **In the end, this comparable will have its price increase by 12.05% if we were to compare it with a target having a volume of 1 000.**

- **Its price goes from 7.70€ to 8.63€. The volume adjusted price increase is 0.93€.**

This volume adjustment is applied on every comparable of every target cpc, no adjustment is applied on the target directly !

6.Group adjustment

The final price of a comparable cpc is going through both the steps of volume adjustment and group adjustment. Which is way we have two intermediate lines displayed in the WebApp.

Original price + volume adjustment impact (section 5) + group adjustment impact (section 6) = volume adjusted price

Group adjustment is not applied for all families (only 13 of the 16 right now). It can change from one run to another depending on the group volumes analysis (explained below).

- We do not currently apply group adjustment for these 3 families :
  - Kalix
  - Ixef
  - Halar

Group adjustment steps :

We mentioned before the "cpc_volume_log" feature which is showing the volume of a cpc. We also have a feature named "group_volume_but_cpc_label" looking at the total volume of the customer group of the cpc, **excluding volume of the cpc itself.**

Indeed, the size of the entire group is supposed to have an impact on the cpc price. That is why this group adjustment step is needed.

Example of the variable SHAP values from the section 3.2 "Features importance : Shap value" :

We can see that like regular volumes, the higher the value is (red ones), the lower the price is.

This "**group_volume_but_cpc_label**" variable has 5 modalities :

- 0_one_cpc : The volume of the group for this family excluding the current CPC is equal to **zero**.
  This means that the group of the customer **only buys this CPC** and the adjustment needed is entirely covered by the step presented in section 5.

- Other CPCs whose groups have sales in the family (excluding the CPC ones) are split in 4 bins representing **quartiles** :
  - 1_small : 25% of the cpc having the smallest group volumes (first quartile)
  - 2_medium : 25% of the cpc representing the second quartile
  - 3_big : 25% of the cpc representing the third quartile
  - 4_top: 25% of the cpc having the biggest group volumes (fourth quartile)

Once every CPC is placed in its modality, we compute the median SHAP value of "group_volume_but_cpc_label" for every modality. This gives us the estimated impact of the group volume on price.

group_volume_but_cpc_label	median_shap
0_one_cpc	0.013
1_small	0.014
2_medium	0.009
3_big	0.002
4_top	-0.036

Example for Amodel

As a reminder, these values come from the SHAP values and therefore are on the **log scale**. After transformation to get out of the log scale, they will be applied as a factor to previously computed volume adjusted price for the families having the group adjustment applied.

Note : this is why the volume adjusted price displayed in the dashboard will not be the same as the one in the WebApp used to define a target cpc price. In the dashboard, we only go through the volume adjustment step to explain it more in-depth, but the group adjustment is not applied.

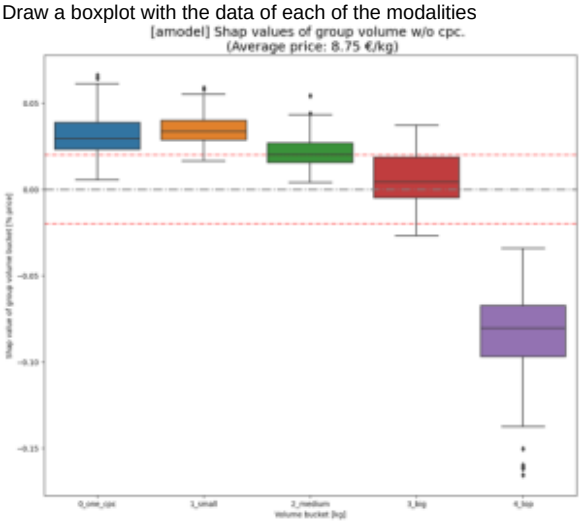
How to define which family has a group adjustment applied ?

To decide if we will apply a group volume adjustment for this family, we :

- Apply the transformation on SHAP values to get out of the log scale and obtain a price variation percentage.

group_volume_but_cpc_label	median_shap
0_one_cpc	0.029
1_small	0.033
2_medium	0.020
3_big	0.004
4_top	-0.080

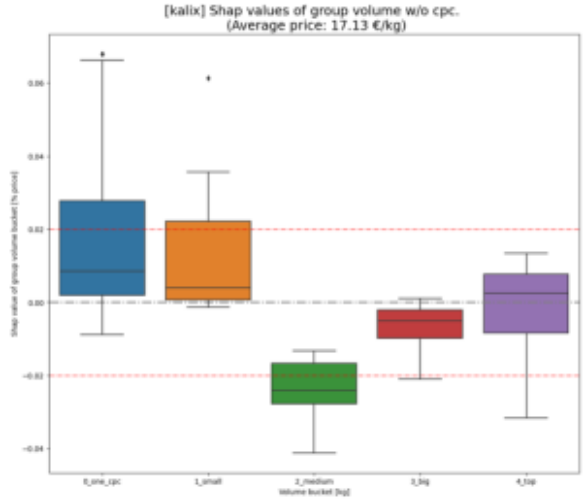
- Draw a boxplot with the data of each of the modalities



This boxplot shows us that customers belonging to small groups are buying at an average of 3% higher than the rest and customers belonging to the biggest groups 8% lower.

Red lines represent the -2% and +2% price variation due to group volume threshold. **We only apply a group volume adjustment when the medians of the different modalities are outside these bounds and the behavior of decreased price is coherent between groups.**

Another example for Kalix :



- The median of one of the groups only is outside the bounds.
- The price variation is not coherent : prices increase for the biggest groups when we expect a decrease.
- **Conclusion : we do not apply a group volume adjustment for this family.**

Once we have the price variation due to group adjustment for our comaprable cpc, we check the difference with our target cpc :

- **If they are both in the same modality, therefore having the same group size considered, we do not apply any adjustment.**
- **If they have different modalities, the adjustment will be the difference between the target bucket and the comparable bucket.**

7.Price recommendation

In the end, the price recommendation is the median price of the 10 closest comparables of the target cpc.

- The comparables price used is the adjusted one including both volume adjustment and group adjustment steps.

The maximum increase is capped at 30%.