Accelerate Clustering

Version 1.0.0

August 2023

Accelerate Clustering

The Clustering Accelerator provides an easy way to create clusters in order to enrich data by creating data-driven labels on any dimensions of the transactions.

Overview (Clustering)

Purpose

Clustering, specially clustering of customers, can be a powerful tool for businesses looking to better understand and target their pricing. By identifying meaningful clusters, businesses can develop more effective pricing and marketing strategies, increase customer satisfaction and retention, and ultimately drive revenue growth.

This Clustering Accelerator intends to provide **a way to group items** using a fairly simple process, not requiring a data scientist to proceed.

Data often comes too sparse or too granular to really leverage their full potential and be actionable, specially when setting a **pricing strategy** that can be handled in a **nice and manageable way**. So using clustering can be really valuable for grouping customers, products, point of sales or any of the pertinent dimensions of your transactions, and this clustering technique enables you to regroup items within a category depending on what happens in another attribute.

The **initial intention** here is to better understand the **relationships** between **customers** and **products**. The clustering idea is to **regroup customers** that buy the same products in the same proportions to create a **dat a-driven typology of customers**. This data-driven typology combined with the sizing of customers provides an accurate understanding of who buys what and allows you to adjust the pricing accordingly.

Versatility of this approach may also help in **regrouping products** with regard to who buys them, or where they are bought, or when etc. Any pair of transactions' dimensions can be explored to build a new dimension that **enriches** the transactions with data-driven labels that help in defining pertinent pricing strategies.

Pricefx Solution

Our clustering model is dedicated to enriching transactional data. To operate it, you need to define the following:

- Grouping dimension to label, called *groups* (e.g. customers)

- Observable dimension to characterize the groups, called *based-ons* (e.g. products or products category)
- Metric

Different metrics serve different purposes, from the spend-pattern analysis to more common statistical metrics. The spend-pattern analysis computes for each group the ratio of the revenue spend for which based-ons, typically product category. As an alternative, statistical metrics can also be used (mean, median, sum) to build clusters as group of customers with similar discounts (then the metric would be "average" discount). Subsequently, the similarity between groups is computed and clustering is applied to result in clusters.

We use a hierarchical clustering algorithm and some additional evaluations to recommend the final number of clusters to the user who can still adjust that number.

To refine the meaning of the clusters, an intermediate analysis helps the user to focus the clustering on the most relevant groups and based-ons with the possibility to label the numerous less relevant groups (i. e. very small customers or long tail products) based on their similarity with already clusterized groups.

Outputs

The output of the model is a list of items grouped into clusters, typically customers grouped by customer segments.

A set of dashboards is also provided in order to review and assess the outputs.

Outputs can be exported directly to a Data Source and joined to a Datamart. Then the clusters can be leveraged to defined pricing strategies or used as a segmentation level in Negotiation Guidance.

Approach

This Clustering Accelerator is based on hierarchical clustering. Hierarchical clustering is a type of unsupervised machine learning algorithm that is used to group similar objects or data points into clusters. The algorithm works by merging the most similar pairs of data points into clusters.

One of the main advantages of hierarchical clustering is that it produces a hierarchy of clusters, which can be then explored to find out the best number of clusters and help users with an optimal set of clusters.

Also, hierarchical clustering is robust to small changes in the dataset, meaning clusters would remain fairly similar over time and should not be affected by small changes of scope.

Additional refinements can be used to define minimum revenue within a cluster or remove less meaningful data (like small customers) before extending the defined clusters by assigning each and every case to a cluster. This is part of the "extended cluster affection".

Limitations

- **Minimal number of items to cluster** – Even though it is possible to make this model execute with a minimum number of 7 items, the clustering result might not be as relevant. As a rule of thumb, the number of items should exceed 10 times the expected number of clusters.
- **Metric required** – The clustering approach relies on a metric (4 ways offered for now) that will be used to find out some pattern in the data and group together the items. So using such metric is a prerequisite and this metric should be defined and can be computed at the right level: granularity group attribute x based on attribute (e.g. Customer x Product Group).

- **No predefined extension point** – There is no out-of-the-box extension point defined for now. If you intend to use your own metric, custom code should be written. (But then the accelerator becomes specific so it cannot be updated without extra effort to port those modifications.)

- **Data requirements** - See Data Requirements (Clustering).

Business User Reference (Clustering)

- Usage (Clustering)

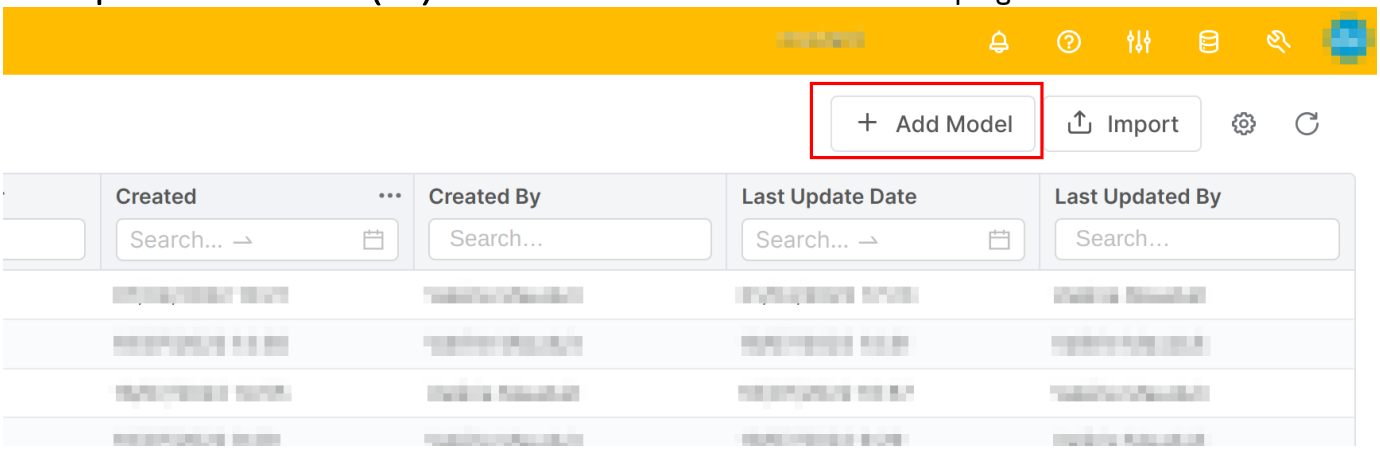
Usage (Clustering)

Take the following steps to configure a model:

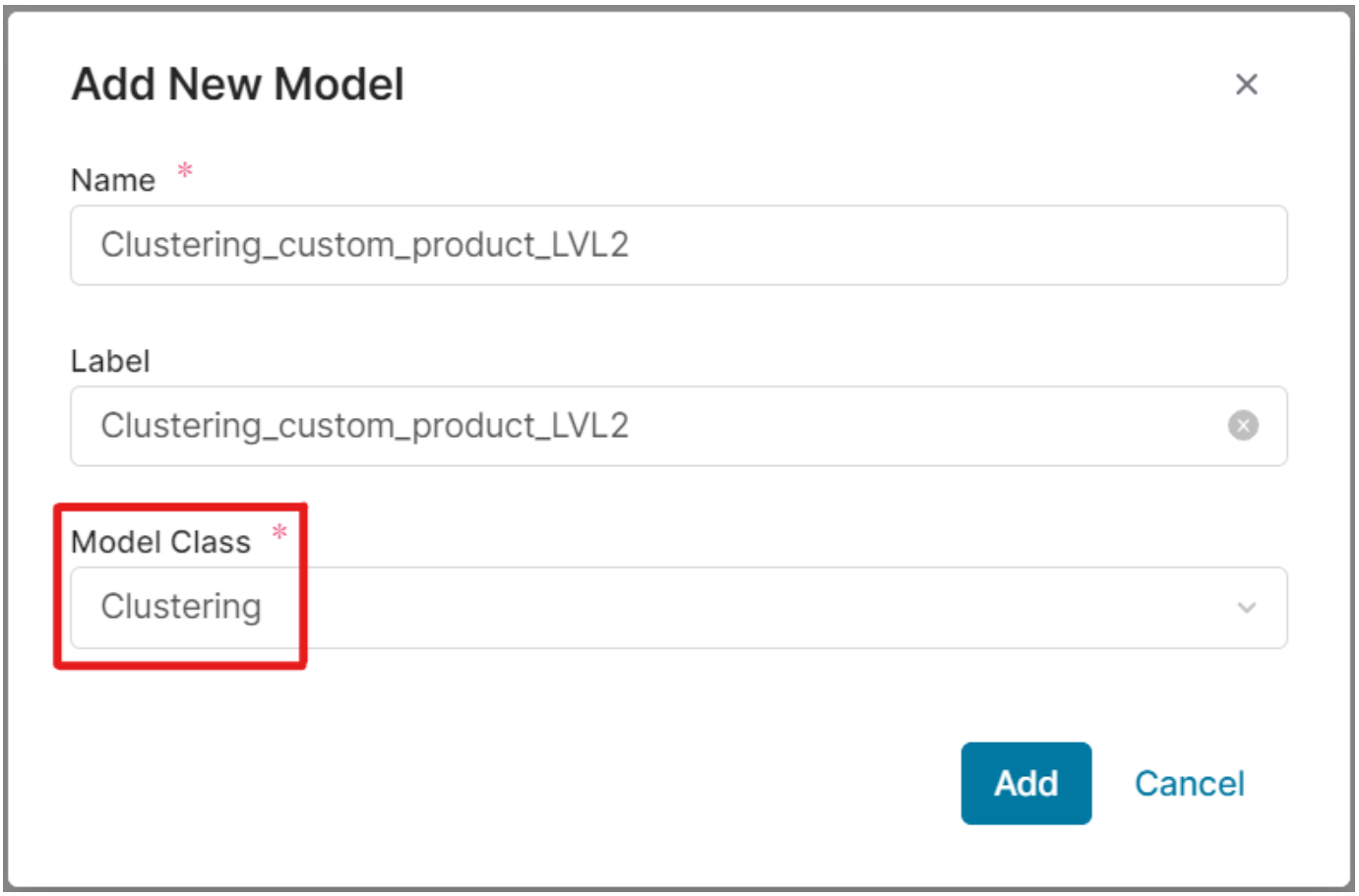
- 1. Create a Model based on Clustering Model Class
- 2. Set the Scope of Transactions (Definition Step)
- 3. Data Overview and Setting Clustering Parameters (Model Configuration Step)
- 4. Explore Clustering Results (Result Step)
- 5. Export Clustering to Data Source
- Additional information

1. Create a Model based on Clustering Model Class

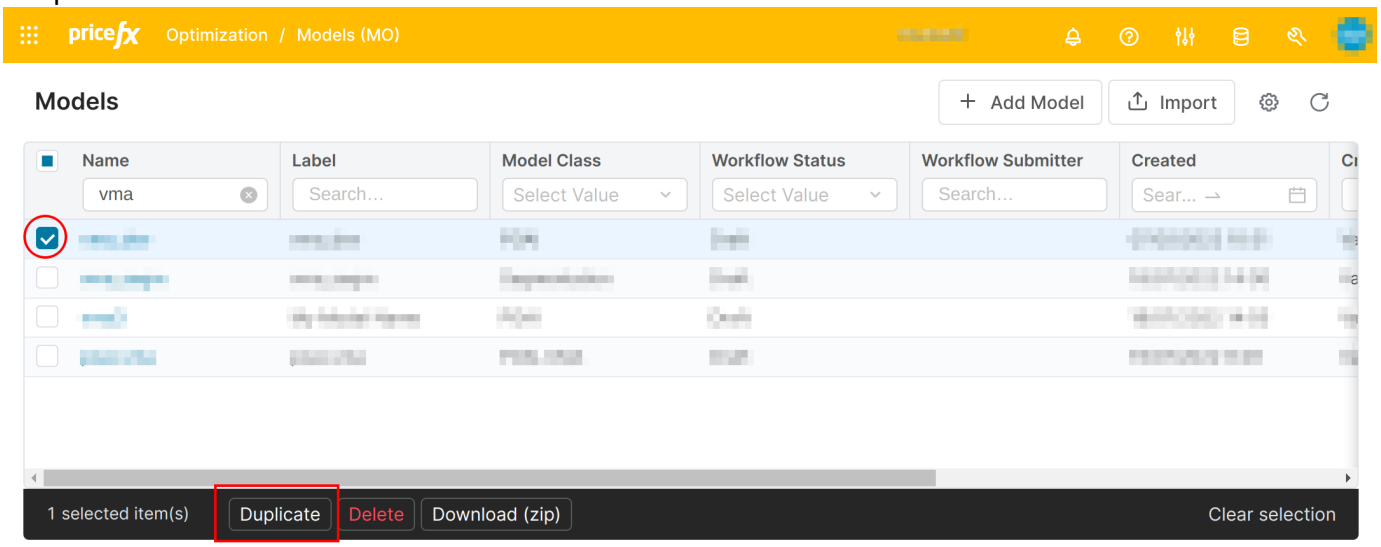
Go to **Optimization** > **Models (MO)** and click the **Add Model** button at the top right.



A pop-up is shown where you provide a name for your model, and the Model Class, which is *Clustering*. Another Model Class would belong to another kind of optimization model.



You can also duplicate an existing model. In this case, you will keep all the inputs of the previous model and you will have to rerun all the steps to get the outputs. Once you have copied a model, you can change its name by double-clicking the blank side of its name/label.
⚠️ Remember to do it before running the model. You cannot change a model name once it has been computed.



The same model class, *Clustering*, can be used by many models. Use informative names for your models, providing information on your dataset, and your calculation case.

2. Set the Scope of Transactions (Definition Step)

In the Definition step, you map the inputs and set the scope of the model. The user inputs are always on the left. Refer also to Data Requirements (Clustering).

- **Source** – Datamart (typically transactions Datamart) used to perform clustering. It must fulfill the requirements listed in Installation (Clustering). Once provided, some fields based on it appear:
  - **Metric** – Select the type of metric that will be the basis of comparison of the items to be clustered. The clustering approach will group together items with similar values.
    - **Spend Pattern** computes the share of revenue (set in the Revenue field) spent across categories (typically product category). A typical use case is profiling the customers based on their consumption, intending to group together customer purchasing similar types of products.
    - **Average** takes the average of the target
    - **Median** takes the median of the target.
    - **Sum** takes the sum of the target.
  - **Target** – Defines the attribute used by the metric computation (Spend pattern or Average/Median /Sum).
    - For the *Spend Pattern* metric, revenue shall be defined as the target, so the share of revenue are compared to define the clusters.
    - For *Average/Median*, discount rate or margin rate can be used for example. So the clustering will group together based on a similar pattern of discount rate or margin rate.
    - For *Sum,* a metric that can be summed can be used, such as absolute value of profit (in that case clustering will be based on the pattern of the generated profit for each case of "group" and "based on").
  - **Group** – Defines the attribute intended to be grouped together, e.g. customers to be grouped in customer clusters. It is the first of the two dimensions used to aggregate the data.
  - **Based On** – Defines the attribute to which the metric will be compared to. This is the second of the two dimensions used to aggregate the data.
    ⚠️ Attributes for "Group" and "Based On" cannot be part of the same hierarchy, e.g. Product Category and Product Sub-Category. In that case the clustering process will fail as no pattern can be found.
  - **Revenue** – Provides the transaction revenue, which will be the basis of some analysis. It may be a net price, gross price, or another, depending on the properties you want to explore.
  - **Additional Features** – Some non-mandatory fields you may want to keep in the data for a further filtering during the step Model Configuration or for making the result review easier.
  - **Advanced Filter** – Allows you to filter the meaningless data, using e.g. time frame (last 12 months is a good start). Nevertheless, use these filters wisely: even though it is possible to make this model execute with a minimum number of 7 "Group" items, the clustering result might not be so relevant. As a rule of thumb, the number of items should exceed 10 times the expected number of clusters. We recommend that you define at least these filters for data cleanliness and thus avoid errors later on:
    - Revenue > 0
    - Quantity > 0
    - Removing null values for "Group" and "Based On"

Once you apply the settings, the right panel provides:

- **Data selection sample** – Filtered out data that will be the scope of the clustering.

3. Data Overview and Setting Clustering Parameters (Model Configuration Step)

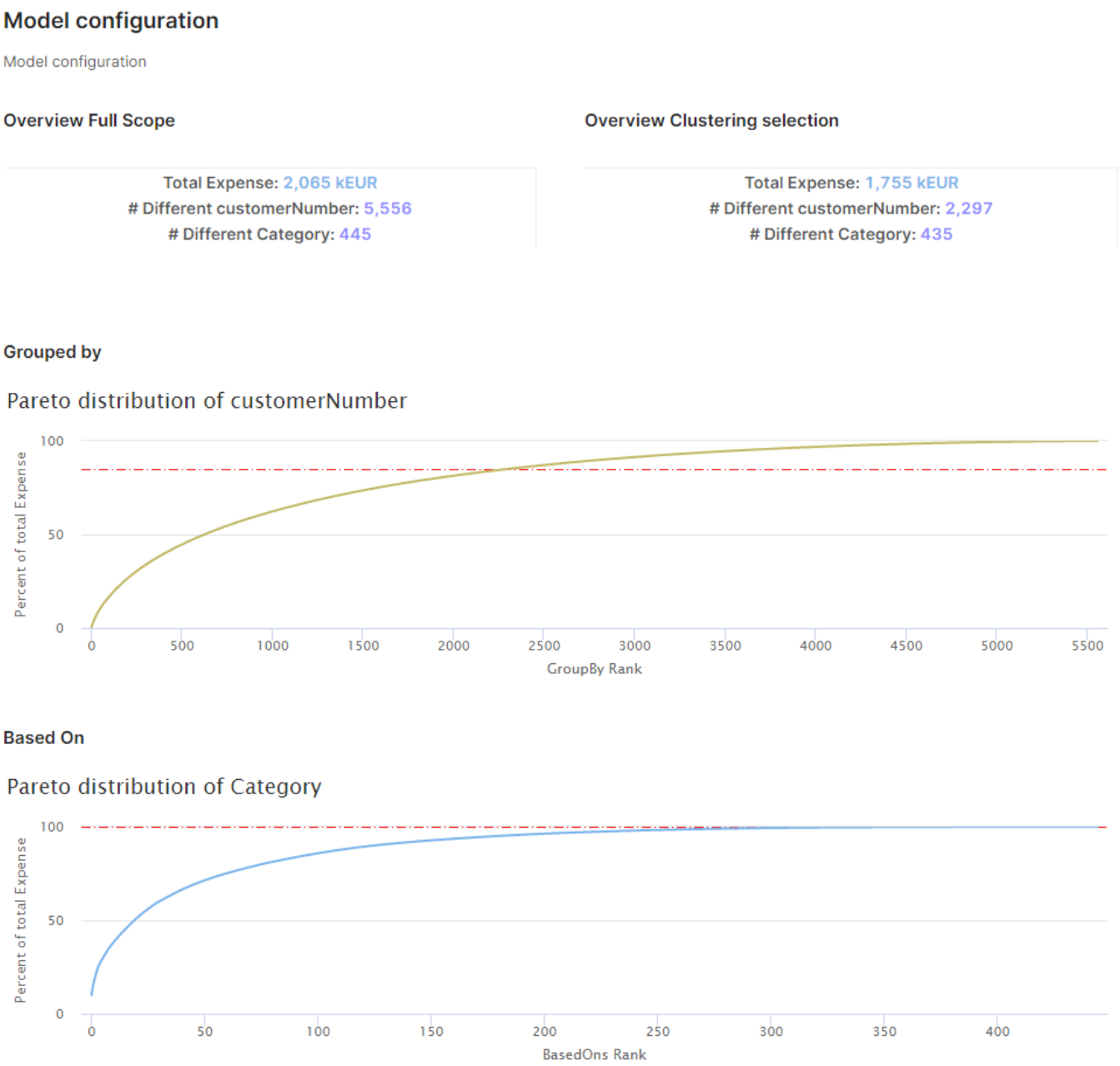
Opening the Model Configuration step triggers an initial calculation run to prepare the data and provide the user with some insights about items to *group* and *based-on* items. Some Pareto distribution diagrams can help the user in setting a threshold to ignore e.g. some very small customers or products that may degrade the clustering process (including less relevant data) and increase computation time.

The user inputs on the left allow you to tune both the clustering process and post-treatment labelling. The clustering process is based on a n*on-supervised hierarchical clustering algorithm* that produces a tree based on the distances between groups. Once this tree is computed, it is possible to select the level of the tree that will be kept as the clustering threshold. Close to the root of the tree means few low discriminant clusters, while close to the leaves means many high discriminant clusters.

- **Minimum Number of Clusters** – Minimum number of clusters among which clusters will be suggested.
- **Maximum Number of Clusters** – Maximum number of clusters among which clusters will be suggested.
- **Minimum Revenue in a Clusters** – Sets the minimum total revenue a cluster should reach. This would remove clusters representing a low total revenue. Any item initially assigned to such a small cluster will then be reassigned to the closest cluster.
- **Percent of GroupBys to process** – Represents a threshold that will keep in the initial clustering process the *group* items that represent together x% of the total revenue. This is a high pass filter that may e.g. remove one-shot customers from the analysis. Remember to keep enough "GroupBys" so that they exceed 10 times the expected number of clusters.
- **Percent of BasedOns to process** – Represents a threshold that will keep in the initial clustering process the *based-on* items that represent together x% of the total revenue. This is a high pass filter that may e.g. remove long tail products from the analysis.
- **Expense percent threshold** – Affects the values in the *group * based-on* matrix in the Expense Pattern analysis: the default value of 1 means that when a group has expended less than 1% of its revenue on a given based-on, this expense is nullified to focus the comparison of groups on the main based-on they are linked to.
- **Linkage method** – Defines the way the clusters are aggregated together for the final outputs. The options to select from are: *single*, *complete*, *average*, *weighted*, *centroid*, ***ward*** (which is by default a good choice). See Linkage method for further details.
- **Name Prefix Clusters** – Non-mandatory string that will be appended as a prefix to the automatically generated names of the clusters. Automatic naming is a summary of the 3 main based-ons that are present in a cluster.
- **Cluster affectation** – Is either *Basic* or *Extended*:
  - **Basic** – Only clusters the *group* items selected for clustering after reaching the thresholds defined above.
  - **Extended** (default) – All group items receive a cluster label; the ones that were not part of the initial clustering process are assigned to existing clusters in the second pass based on their proximity to already clustered group items.
- **Show detailed heatmap in result** – If set to Yes, this parameter visualizes all or a sample of the group and how the groups are put together in clusters.
- **Suffix for Data Source** – String that is added to customize the name of the final exported Data Source.

In order to apply any changed parameter, it is necessary to click the **Apply Setting** button at the bottom left of the panel.

This button will also update the right panel values to provide an estimation of the eventual filtered out group items and based-on items.



When all parameters are correctly set, click the top right **Continue** button to trigger the clustering process.

4. Explore Clustering Results (Result Step)

When you arrive at the Result step from the Configuration step, the model runs a calculation that can take some minutes, depending on the size of the data and the number of *groups* and *based-ons*.

Once the calculation has run, three tabs appear:

- **Overview** – Exposes the best clustering occurrence that respects the user's settings.
- **Details (groupBy)** – Lists all the groups and how they have been assigned to a given cluster.
- **Number of clusters** – Shows many clustering alternatives based on a single linkage matrix.

4.1 Overview

This tab has 6 widgets:

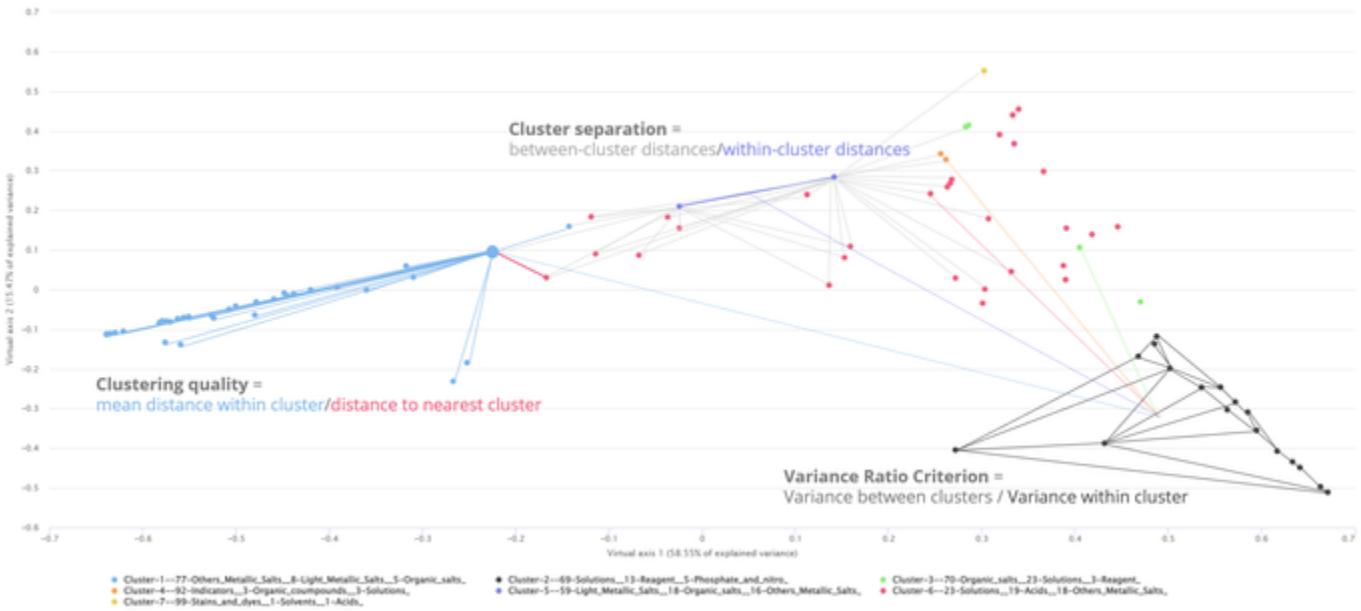
Clusters

List of clusters with their name (*ClusterName*) that describes the main composition of the cluster, number of groups (*Nb_+"name of column of the group"*), number of transactions (*Nb_Transactions*), and the total expense in the cluster (*TotalExpense*). *NumberOfSizes,* most often set to 5, shows the number of quintiles that are computable inside a cluster. If is is less than 5, it means that some groups are relatively very big compared to others in this cluster. *TotalExpenseRatio* is the percent of expense/revenue content in a cluster.

Clustering Numbers

Some high-level information about the current clustering solution.

Clustering Metrics



For description of the 3 metrics used for comparing clustering solutions see Clustering Metrics (Clustering).

Relative Expenses per Cluster

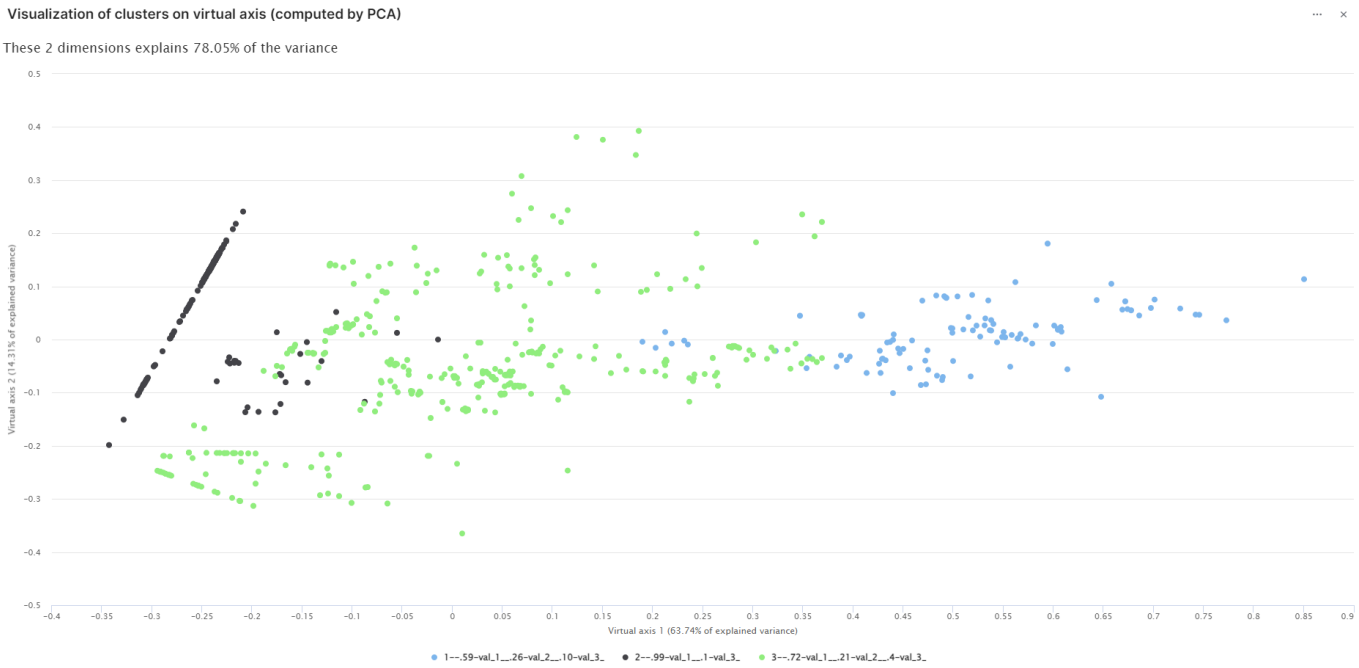
Heatmaps that present the average behavior of each cluster according to the based-on dimensions. The data are normalized between 0 and 1.

Details of Clusters

Information on the clustered groups, such as the cluster the group belongs to, the relative size of this group inside the cluster (A = big, E = small), the rank of the group for a cumulative revenue point of view, the rank of the group for a number of transaction point of view etc.

Visualization of Clusters on Virtual Axis (Computed by PCA)

This is a way to visualize all the values how they have been assigned to a cluster, with a two dimensional plot that has been generated for that purpose. Axes are produced by a reduction of dimensionality of all the based-ons and the two axes of the plot are the two main axes to split apart all the data. The percentage of explained variance computed on the *group x based-on* matrix is displayed for each axis and in the subtitle. Groups are colored according to the cluster label they have received.



4.2. Details (groupBy)

Details of Clusters

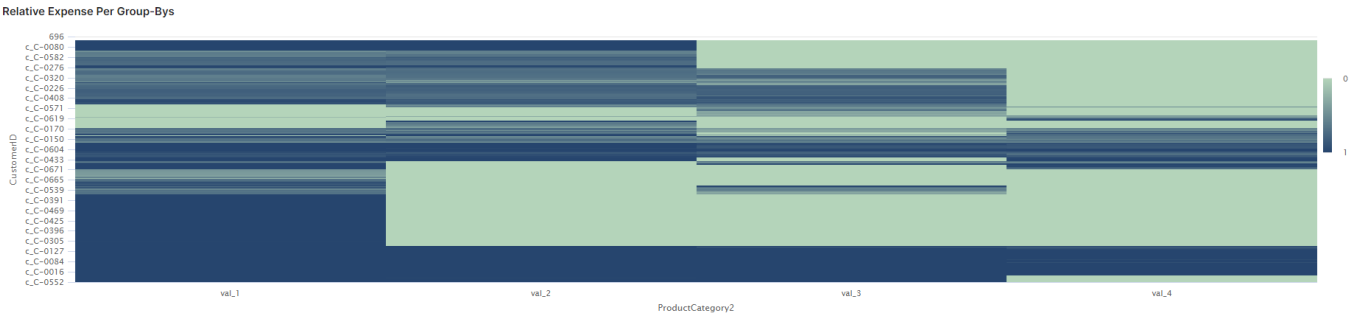
Information on the clustered groups, such as the cluster the group belongs to, the relative size of this group inside the cluster (A = big, E = small), the rank of the group for a cumulative revenue point of view, the rank of the group for a number of transaction point of view etc.

Details of Target Metric

Indexed matrix *group x based-on*.

Relative Expense Per Group-Bys

Heatmap made from indexed *group x based-on* matrices. This rendering requires a lot of resources to be shown properly, hence some sampling of the groups can be done. This visualization presents similar groups and their proximity.



If the heatmap happens to have too many cells, only a subset is displayed.

4.3. Number of Clusters

Clustering Absolute Metrics (Higher Is Better)

Raw metrics values (non scaled) can be useful for the comparison of the results of different models.

Clustering Absolute Metrics (higher is better)

With 3 clusters

Perfect quality score is 1.

Clustering quality: 0.338

Perfect separation score is 1.

Cluster separation score: 0.49

From 0 (worst) to high value (good), influenced by number of attributes

Variance Ratio Criterion: 424.6

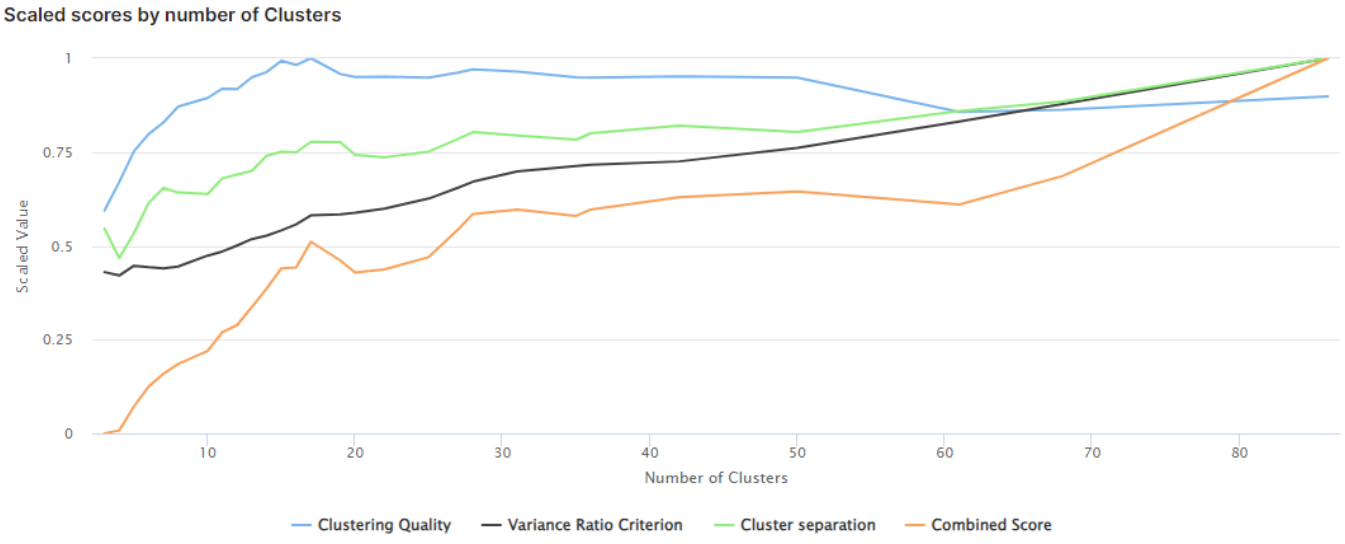
Impact of thresholds:

0 clusters below revenue threshold and concerning 0 CustomerID

Further details about the score: Clustering Metrics (Clustering)

Scaled Scores by Number of Clusters

This is a visualization of the metrics across all the clustering options offered by the hierarchical tree built behind the scene and before any threshold. It is helpful to assess if the range of the min-max numbers of clusters set in the Configuration step is well defined or should be adjusted. Nevertheless, you should prefer a reasonable number of clusters rather than some good metrics and an impractical number of clusters.



Clustering Alternatives

This table contains high level data about the different clustering solutions of the same hierarchical tree, including all scores for all number of clusters.

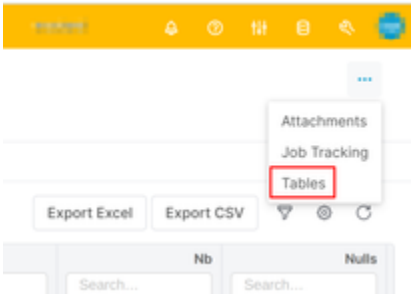
5. Export Clustering to Data Source

Clicking the **Continue** button from the Result step triggers the export of the model table clusterd_groupbys to a Data Source that will be named:
*CLUST_[name of **group**]_[name of **based-on**]_[name of **target**]_[optional suffix]*

All names are limited in length to 7 characters, and if a Data Source already exists with the same name, this old Data Source will be overridden.

Additional information

All computation results are stored as tables of the model. These tables can be accessed through the menu in the top right corner. But usually you do not need to access them this way; all needed information is directly provided in the three result sections of the model.



Admin User Reference (Clustering)

- Installation (Clustering)

Installation (Clustering)

The Clustering Accelerator deploys the Clustering Model Class and the related logics.

Prerequisites

Before you start the installation of the Accelerator, ensure you have the according *Transactions Datamart*. Some important points about it:

- It must contain the required fields – create them if needed:
  - **Customer field** – ID of each customer. Often used as **group by** later in the clustering. If you want to clusterize other items than customers, make sure you have these fields in the Datamart.
  - **Product field** – ID of each product/article/SKU. Often used as **based on** later in the clustering. If you want to characterize clusters based on other items (and not products), make sure you have these fields in the Datamart.
  - **Revenue field** – Extended price of the transaction row.
  - **Target field** – It must be a numeric dimension; generally, it is a revenue, margin, margin rate (margin /revenue), or a discount rate (discount/revenue).
- Avoid null values in **Group by** and **Based on** dimensions. If necessary, create a new field by using an expression that replaces empty values with a text like "Not provided".

Deployment

1. Access PlatformManager at https://platform.pricefx.com/ and log in with your account or using O365.
2. Go to **Marketplace** > **Accelerator Packages**.
3. Find **Optimization - Clustering**.
4. Select your Target Partition from the drop-down menu.
5. Click **Continue** and wait until the deployment is complete.

Technical User Reference (Clustering)

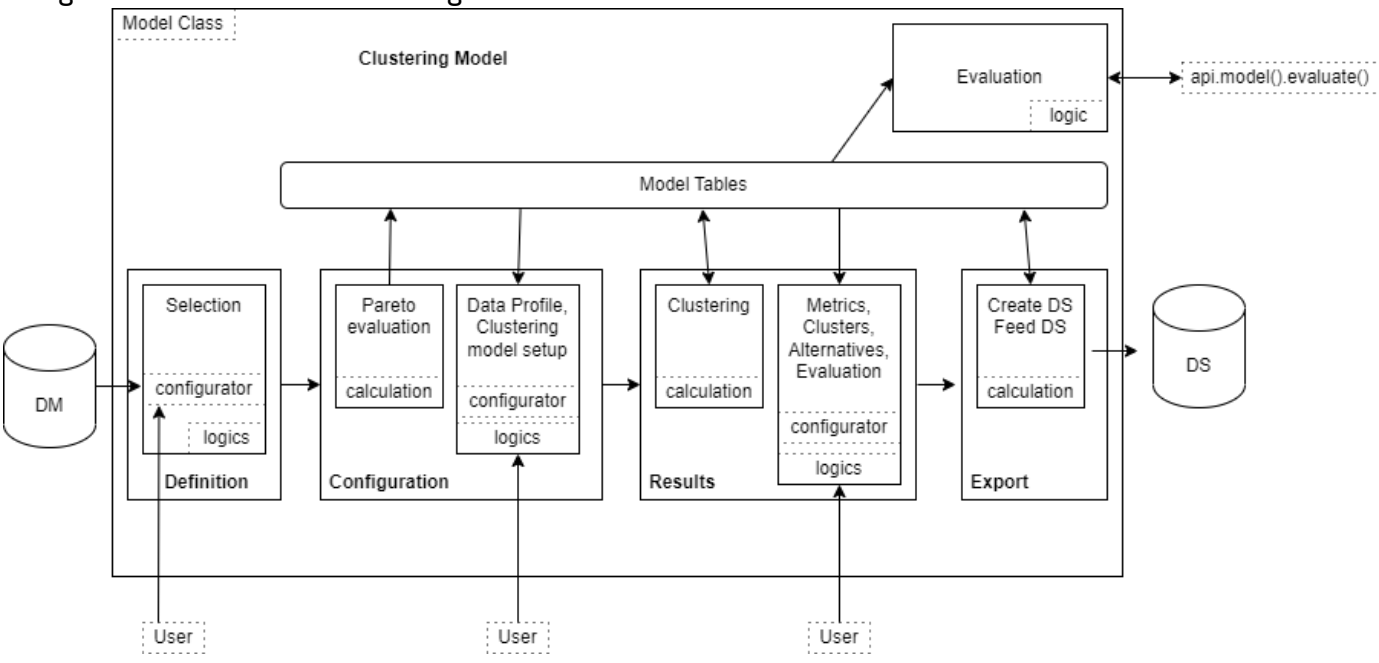
This section details the ModelClass and logics that the Clustering Accelerator deploys. For each step, its aim, outputs, and the main reasons to modify the logics are explained.

In this section:

Clustering Model Class

The Clustering ModelClass organizes a list of logics to create the model architecture. It is a JSON file that refers to some logics and it is transformed into an optimized UI in the Pricefx platform.

The general architecture of the Negotiation Guidance Model Class is:



It defines four steps

- **Definition** – Sets the scope of the transactions.
- **Configuration** – Sets the parameters for the clustering.
- **Results** – Looks at the outputs of the clustering, the properties of the clusters and eventually how to change the configuration toward a better clustering.
- **Export** – Copies the result of the clustering into a Data Source.

Library

The logic is **Clustering_Library**.

Aim of the logic

Clustering_Library is used in nearly all the other logics deployed by the Accelerator and defines a set of functions needed specifically for this Accelerator, but also some constants used to easily change the user interface wording. There are the following elements:

- `Parameters` – Contains a function to check the type of the columns when exporting some tables.
- `Configurators` – Groups the methods to deal with formatting user inputs.

It is accessed via the calls on `libs.Clustering_Library.XXX` in the code.

Common reasons to modify the logic

If there is another kind of input to deal with.

Definition Step

There is no calculation logic in this step, and there is one tab with related evaluation logics: **Clustering_definition_eval** and **Clustering_defnition_eval_configurator**.

Aim of the logics

These logics provide the user inputs to define the source data and map it, and to define what kind of metric will be used later in the clustering process.

Outputs of the evaluation

A table of the filtered transactional data that will be used for the clustering.

Common reasons to modify the logics

- Some other mappings are needed or would be retrieved.
- Some customized metrics that will require specific developments.
- Define pre-set filters.
- Add a chart to better understand the data. (Caution: it can take long, as the data are not yet stored in the model.)

Model Configuration Step

Contains one calculation logic **Clustering_configuration_calc** that executes when accessing this step and one tab split in two panels, one for user inputs, the other for evaluation.

Calculation: PriceDrivers

The logic is **Clustering_configuration_calc**.

Aim of the logic

Aggregates the transactions at the levels selected by the user in the Definition step and computes the Pareto distributions of the two levels selected.

- Outputs of the calculation
  - Model tables of the aggregated data.
- Common reasons to modify the logic
  - Add a chart to better understand the data.

Setup Panel

The logic is **Clustering_configuration_configurator**.

- Aim of the logic
  It collects user's inputs for:
  - Reducing the scope of the clustering to the main items based on their Pareto's ranking.
  - Setting clustering algorithm parameters (linkage metric, number of clusters...).
  - Setting post processing parameters (extension on cluster affectation to out of scope items).
  - Customizing clusters' names and Data Source's name.
- Outputs of the evaluation
  User inputs.
- Common reasons to modify the logic
  There could be different parameters to customize the clustering or eventually you can add other clustering algorithms.

EvaluationPanel

The logic is **Clustering_configuration_dimensions**.

- Aim of the logic
  Exposes to the user some information about the final scope of the clustering:
  - Pareto distribution of the items to be clustered (**group-by**).
  - Pareto distribution of the items used for qualifying the clusters (**based-on**).
  - Scope impact of thresholds set by the user.
- Outputs of the evaluation
  Displayed Pareto diagrams.

  Overview before/after the scope setting.
- Common reasons to modify the logic
  There could be different parameters to customize the scope of the clustering and their impact should be exposed here.

Result Step

Contains one calculation logic **Clustering_clustering_calc** that is executed when accessing this step and one tab split in four tabs: Overview, Details (Group by), and Details.

Overview

The logics are **Clustering_result_matrix** and **Clustering_result_pca**.

- Aim of the logics

This dashboard provides the results of the clustering based on the thresholds in some meaningful portlets. It presents the data of the clusters, how they are composed, the business they represent and how different/close they are (see PCA plot).

- Outputs of the evaluation
  Like any evaluation logic, there is no real output, it displays a dashboard.
- Common reasons to modify the logics
  To display other charts or to provide meaningful information in a different way.

Details (Group by)

The logic is **Clustering_expense_matrix**.

- Aim of the logic
  This dashboard provides the results of the clustering through the clustered items (**group-by**). If the cardinality of the product **group-by** per **based-on** is low enough, a heatmap shows the group-bys ordered by the clusters they have been assigned to.
- Outputs of the evaluation
  Like any evaluation logic, there is no real output, it displays a dashboard.
- Common reasons to modify the logic
  To display other charts or to provide meaningful information in a different way.

Details

The logic is **Clustering_metrics**.

- Aim of the logic
  This dashboard provides clustering alternatives that are explored by adjusting the maximum distance between clustered items. The clustering produces a hierarchical tree, and adjusting the distance parameter allows you to select a growing number of clusters as you go from the trunk to the leaves.
- Outputs of the evaluation
  Like any evaluation logic, there is no real output, it displays a dashboard.
- Common reasons to modify the logic
  To display other charts or to provide meaningful information in a different way.

Export Step

Contains two calculation logics **Clustering_export_createDS** and **Clustering_export_feedDS.**

- Aim of the logics
  They simply copy the main model table to a Data Source whose name is customizable by the user.
- Outputs of the evaluation
  A Data Source.
- Common reasons to modify the logics
  Either export other tables or suppress this one.

Evaluations

The model has one evaluation: `Clustering_api_CustomerCluster`. For more details about model evaluations see https://pricefx.atlassian.net/wiki/spaces/KB/pages/3851026646/Query+Optimization+Engine+Results#Using-the-Evaluator.

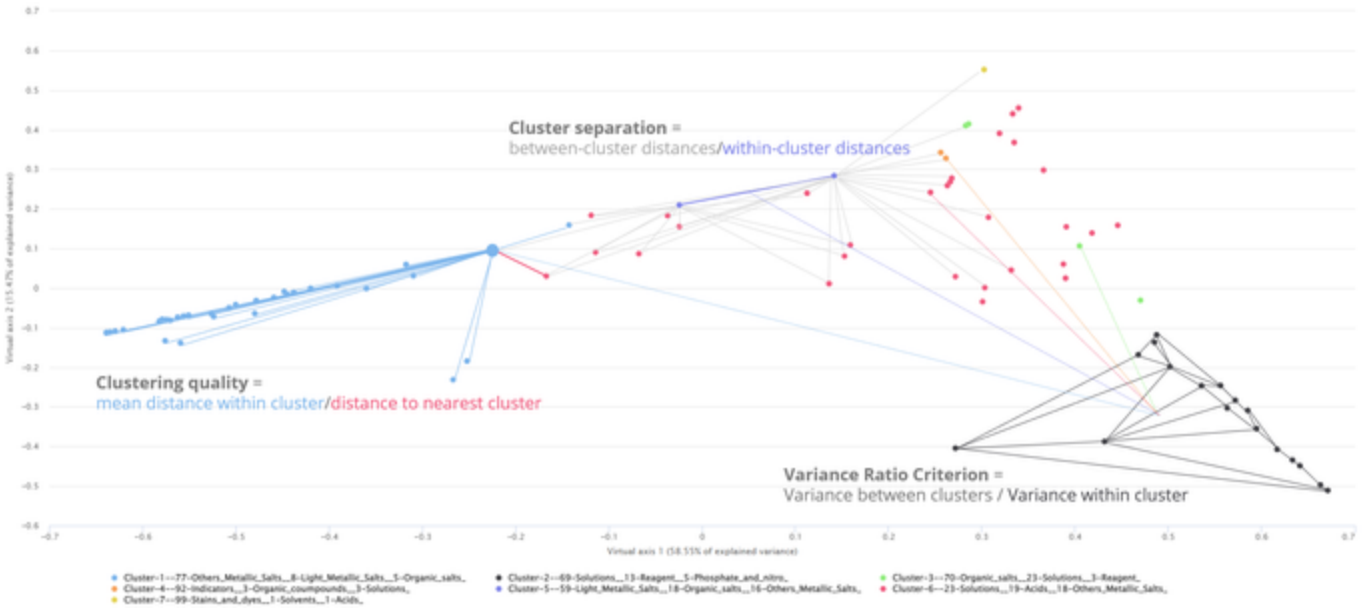
Data Requirements (Clustering)

Clustering operates on a Transactions Datamart or possibly Datasource that has to contain the correct data. Four columns at least have to be prepared for their expected role:

- **Revenue** - Must be a numerical type or money type column without missing or negative values in order to have correct values displayed in the results.
- **Target** - This is the metric that will be used for the clustering itself. It must be a numerical type or money type column without missing values for the normal functioning of the clustering process. Target and revenue may point toward the same column in the Datamart, for example if Spent Pattern is used as metric.
- **Group** - Defines the dimension that will be clustered, e.g. if you want to regroup customers into groups. Must be a dimension type column without missing values. For such cases, it is recommended to handle it in advance: replace missing data by a chain of characters without a space, e.g. "unknown" or "missing-data" . At least 2 different groups have to be present in the dataset.
- **Based On** - Defines the dimensions that will be used to compare the groups. It must be a dimension type column without missing values. At least 2 different values of *Based on* have to be present in the dataset. Also the dimension for *Based on* should be different than the one for *Group* and cannot be part of the same hierarchy (to be specific, those two dimensions cannot be collinear).
  It is advised to handle missing values in advance: replace missing data by a chain of characters without a space, e.g. "unknown" or "missing-data".

Clustering Metrics (Clustering)

Visual representation of the clustering metrics:



Clustering Quality

Clustering Quality is calculated from the Silhouette Score:

- A higher value is better.
- The worst is -1, clusters are assigned in the wrong way.
- 0 is still bad, the distance between clusters is not significant
- 1 is perfect, clusters are perfectly apart from each other and clearly distinguished.

The Silhouette Score is defined by the `s = mean((ba)/max(a,b))` where:

- a - The mean distance between a sample and all other points in the same class.
- b - The mean distance between a sample and all other points in the *next nearest cluster*.

Variance Ratio Criterion

Variance Ratio Criterion is calculated from the Calinski-Harabasz Index and represents the ratio of between-clusters dispersion and within-cluster dispersion.

- A higher value is better.
- The worst is 0.
- The number of samples matters, so the value will change a lot between two datasets.
- There is no perfect value.

The Calinski-Harabasz index is the ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters (where dispersion is defined as the sum of distances squared).

Cluster Separation

Cluster Separation is calculated from the Davies-Bouldin Index as:

`"Cluster separation" = 1 / (1 + Davies-Bouldin Index)`

- A higher value is better.
- A perfect value is 1.

The Davies-Bouldin Index relates to a model with better separation between the clusters.

It is computed as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters which are farther apart and less dispersed will result in a better score.

Combined Score

Coming from all the scores combined together after scaling each score for clarity:

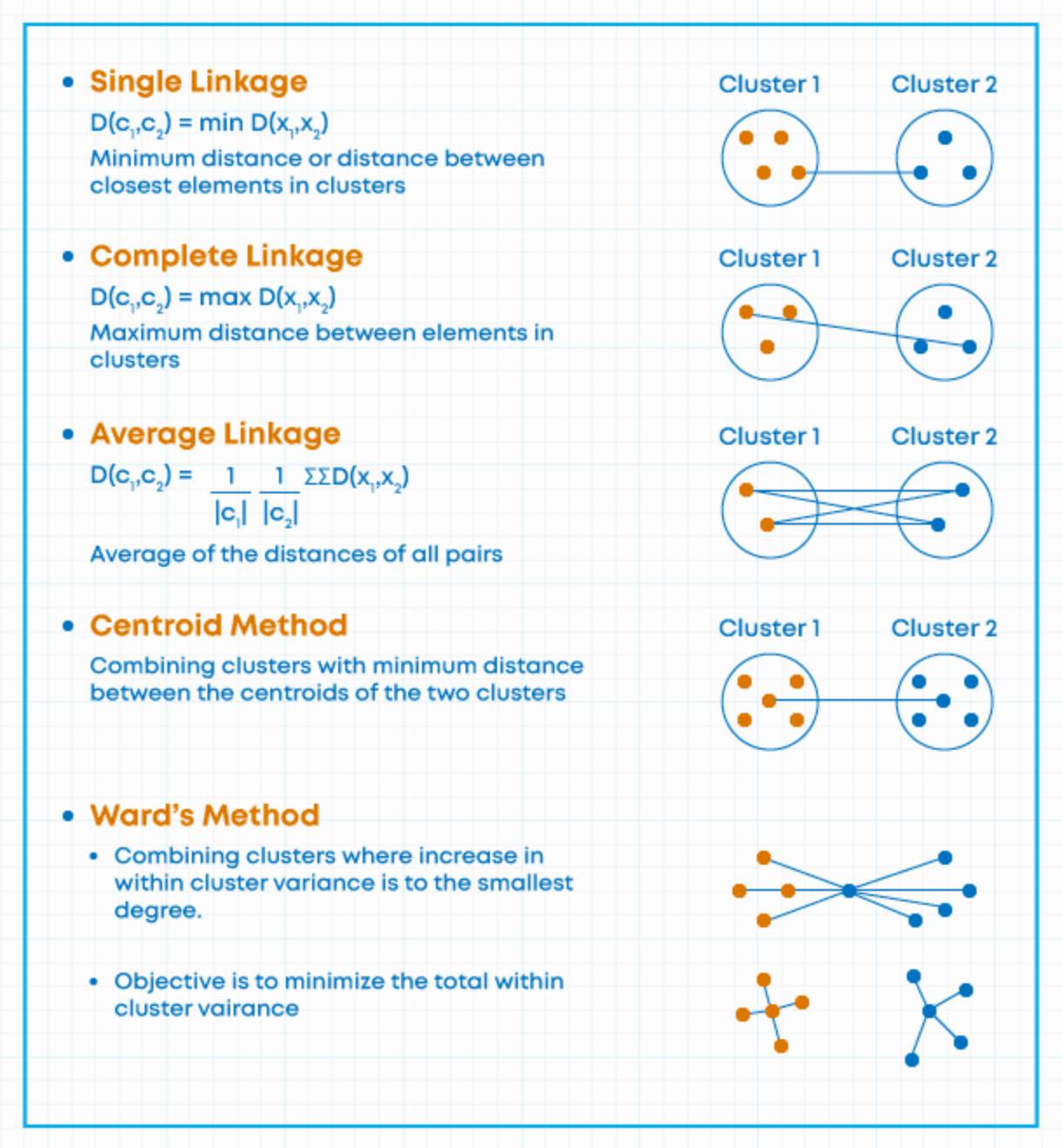
- **Scaled Clustering Quality** = Clustering Quality / Max("Clustering Quality" for all numbers of clusters)
- **Scaled Variance Ratio Criterion** = Variance Ratio Criterion / Max("Variance Ratio Criterion" for all numbers of clusters)
- **Scaled Cluster Separation** = "Cluster Separation" / Max("Cluster Separation" for all number of clusters)

Combined Score = "Scaled Clustering Quality" * "Scaled Clustering Quality" * "Scaled Variance Ratio Criterion" * "Scaled Cluster Separation"

Linkage Methods (Clustering)

The following linkage methods are used in hierarchical clustering:

- **Ward Linkage** (default) – Is based on the idea of minimizing the variance within each cluster. It calculates the distance between two clusters by measuring how much the sum of squared deviations from the mean changes when the two clusters are merged.
- **Single Linkage** – Calculates the distance between two clusters by taking the minimum distance between any two points in the clusters, so it is based on the closest two points in each cluster.
- **Complete Linkage** – Calculates the distance between two clusters by taking the maximum distance between any two points in the clusters, so it is based on the furthest two points in each cluster.
- **Average Linkage** – Calculates the distance between two clusters by taking the average distance between all pairs of points in the clusters.
- **Weighted Linkage** – Variation of average linkage that takes into account the size of each cluster when calculating the distance between them. Larger clusters have a greater influence on the distance calculation than smaller clusters.
- **Centroid Linkage** – Calculates the distance between two clusters by taking the distance between their centroids, or the average position of all points in the cluster.



Source and further explanations: https://dataaspirant.com/hierarchical-clustering-algorithm/#t-1608531820440