MathIAs+® Structural Intelligence Studio — User Documentation

Edition : 1.1 - June 26, 2026

1 Introduction

1.1 Purpose of this document

This document provides a comprehensive user guide for MathIAs+ Structural Intelligence Studio.

It describes how to access the platform, upload structured datasets, run deterministic analyses, interpret results, and manage generated artifacts in an enterprise context.

The platform delivers reproducible and auditable analytical outputs designed to support human-supervised decision-making.

This document is intended for operational use in production environments.

1.2 Intended audience

This guide is designed for:

- Data analysts
- Technical operators
- Decision-makers using analytical outputs
- Enterprise users interacting with the platform

2 Data Science Foundations

2.1 Overview

MathIAs+ Structural Intelligence Studio is built on proprietary analytical methods developed within the Mathias Plus research and development program.

These methods were validated as part of the *R&D Milestone A* on April 30, 2026, and formally registered in France with the INPI on May 1, 2026.

The platform implements deterministic analytical processes designed to produce stable, reproducible, and traceable clustering results in enterprise environments.

2.2 Deterministic Clustering Principles

The analytical approach implemented in the platform is based on deterministic clustering principles applied to structured datasets.

Key characteristics include:

- **Deterministic execution**
  Identical datasets and configurations systematically produce identical results.
- **Structural consistency**
  Analysis is driven by intrinsic data structure rather than probabilistic variation.
- **Reproducibility**
  Results can be regenerated at any time under identical conditions.

This approach differs from stochastic clustering methods that rely on random initialization or probabilistic convergence.

2.3 Structural Analysis Approach

MathIAs+ methods focus on identifying stable structural patterns within datasets, including:

- relationships between variables
- consistent grouping patterns
- invariant segmentation structures

The analytical process is designed to:

- preserve data integrity
- ensure interpretability
- eliminate artificial variability

2.4 Core Proprietary Innovations

The platform integrates two core proprietary innovations developed by Mathias Plus.

2.4.1 MRC Algorithm — Deterministic Recursive Partitioning on K

The MRC (MathIAs+ Recursive Clustering) algorithm introduces a deterministic partitioning approach based on recursive construction over K.

Unlike conventional clustering methods, the partition for *K+1* clusters is derived from the structure of the partition at *K*.

This enables:

- progressive partition refinement
- structural continuity across cluster levels
- elimination of independent re-initialization

In practical terms:

- centroids computed at level K are reused and extended to generate the partition at K+1
- partitions are not computed independently but constructed through a controlled deterministic process

This quasi-refinement approach ensures coherence across segmentation scales.

2.4.2 MPS — MathIAs+ Partition Score

The MPS (MathIAs+ Partition Score) is a proprietary scoring method designed to evaluate the quality of partitions generated by the MRC algorithm.

Unlike conventional clustering metrics, MPS:

- is fully deterministic
- is specifically designed for partitions generated through MRC
- operates on the combinatorial structure of cluster pairs

The score is computed through a deterministic evaluation of relationships between clusters, enabling:

- consistent comparison of partitions
- stable evaluation across executions
- direct interpretability in operational contexts

2.5 Execution Model

Analytical processing is performed through controlled execution pipelines.

Each execution:

- operates on a defined dataset
- follows a deterministic transformation logic
- produces traceable outputs

The internal implementation relies on proprietary algorithms optimized for:

- scalability
- stability
- structured data environments

Algorithmic implementation details are not exposed, but the resulting behavior is observable through reproducibility and auditability.

2.6 Output Interpretation

Results generated by the platform:

- reflect stable structural groupings
- do not depend on probabilistic assumptions
- provide a direct basis for segmentation and decision support

Users are not required to:

- tune hyperparameters
- interpret statistical uncertainty

Outputs are designed to be:

- operational
- interpretable
- directly usable

2.7 Positioning and Scope

The analytical approach implemented in MathIAs+ Structural Intelligence Studio is designed for:

- enterprise-scale datasets
- structured analytical environments
- decision-critical workflows

It complements, rather than replaces, exploratory data science.

The platform provides:

- stable analytical baselines
- reproducible segmentation
- auditable outputs

3 Platform Overview

3.1 General description

MathIAs+ Structural Intelligence Studio is a SaaS platform designed to perform deterministic clustering and structured data analysis.

The platform enables users to process datasets, execute repeatable analytical jobs, and generate auditable outputs for decision-making.

3.2 Core concepts

The platform is structured around the following core concepts:

- **Dataset**
  Structured data uploaded by the user for analysis.
- **Job**
  A unit of execution representing one analysis run.
- **Artifact**
  Output files generated by a job (processed data, metrics, results).
- **Dashboard**
  Interface used to visualize and interpret results.

4 Access and Authentication

4.1 Subscription

Access to the platform is provisioned through Microsoft Azure Marketplace.

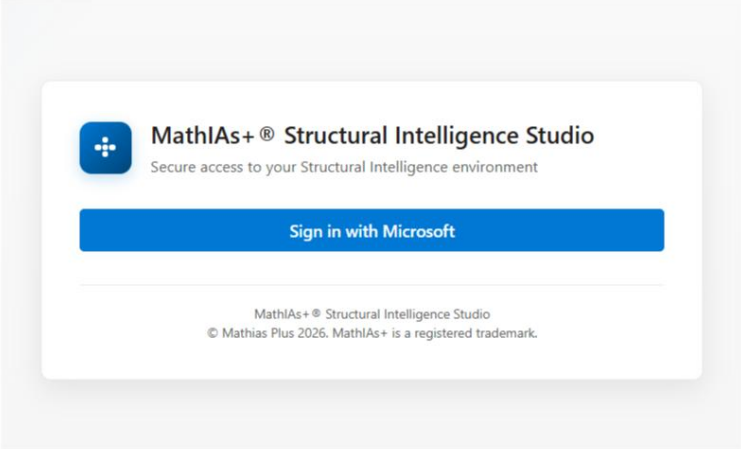
Upon subscription, the service becomes available to the organization.

4.2 Authentication

Users access the platform via:

- Microsoft Entra ID (Single Sign-On)

No local account creation is required.



4.3 User roles

The platform distinguishes three operational roles associated with a subscription:

- **Owner (immutable)**
  The Owner is the user who performed the initial subscription through Azure Marketplace. This role is automatically assigned and cannot be modified.
- **Admin**
  Admin users manage access and assign roles within the organization.
- **Executor**
  Executor users operate the platform by executing analysis jobs and accessing results.

4.4 Execution and billing permissions

All three roles — Owner, Admin, and Executor — are authorized to launch analysis jobs.

Each job execution may trigger usage-based billing (metered billing), depending on the workload executed.

All executions are associated with the subscription context to ensure traceability of usage and billing.

4.5 Access Control (RBAC and whitelist management)

Access to the platform is managed through the **Access Control** interface.

This interface allows subscription administrators to define and maintain a controlled list of authorized users.

Key principles include:

- **Whitelist-based access**
  Only explicitly authorized users can access the platform.
- **Role assignment per user**
  Each user is assigned a role (Admin or Executor) according to the subscription policy.
- **Subscription-level governance**
  Access policies are defined and enforced independently for each subscription.

4.6 Access request workflow

Users belonging to the same Microsoft Entra ID tenant may request access to the platform.

The process is as follows:

1. A user submits an access request through the platform interface
2. The request generates a unique **request_id**
3. The request becomes visible to subscription administrators in the Access Control interface

The requesting user can share the **request_id** internally to support and justify the access request with administrators.

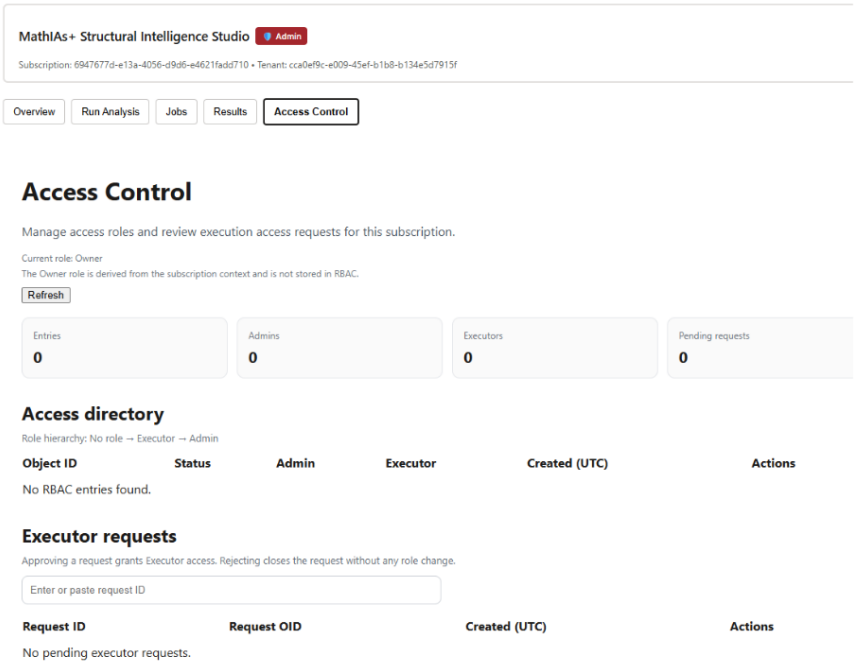
4.7 Validation and assignment

Administrators review incoming requests and may:

- approve the request by assigning a role (typically Executor or Admin)

- reject the request

All access decisions are enforced at the subscription level.



4.8 Security and traceability

This model ensures:

- controlled onboarding of users
- traceability of access requests and decisions
- alignment with organizational access policies

Access decisions are enforced at runtime for all platform operations, including:

- job execution
- artifact access

5 Interface Overview

5.1 Main sections

The platform interface is organized into the following main sections, accessible through the top navigation bar:

- **Overview**
  Provides a summary of the current subscription context and platform status.
- **Run Analysis**
  Allows users to upload datasets, configure parameters, and launch analysis jobs.
- **Jobs**
  Displays the list of submitted jobs, including their execution status and history.
- **Results**
  Provides access to outputs and artifacts generated by completed jobs.
- **Access Control**
  Enables administrators to manage user access, roles, and access requests.

5.2 Navigation logic

The platform does not follow a strictly linear workflow. Instead, users interact with the system through job-based operations and dedicated sections.

A typical usage flow is:

1. Navigate to **Run Analysis** to upload a dataset and launch a job
2. Monitor execution in the **Jobs** section
3. Access outputs in the **Results** section
4. Use **Overview** for general context and system information
5. (Administrators) Manage users and permissions via **Access Control**

5.3 Interface characteristics

The interface is designed around the following principles:

- **Section-based navigation**
  Each function is accessible directly via a dedicated tab.
- **Job-centric workflow**
  All analytical operations are performed through explicit job execution.
- **Separation of concerns**
  Execution (**Run Analysis / Jobs**), results (**Results**), and governance (**Access Control**) are clearly separated.

6 Data Upload

6.1 Supported formats

The platform supports **structured datasets in Parquet format only**.

- **Supported file format:**
    - Parquet (.parquet)

No other file formats are supported.

6.2 Data structure and types

Input datasets must comply with the following specifications:

- **Column types**
  Only numerical data types are supported:
    - integer
    - float
- **Boolean values**
  Boolean variables must be converted to integer format (e.g. 0 / 1) prior to upload.
- **Index column (optional)**
  The first column of the dataset may be used as an index.
  This option can be configured at runtime during analysis setup.
- **Feature names**
  Column names (feature names) are not used in the analytical computations.
  They are only referenced in generated analysis outputs.

For confidentiality purposes, column names may be anonymized without affecting results.

6.3 Upload process

To upload a dataset:

1. Select a Parquet file
2. Upload the file to the platform
3. Confirm dataset availability

Once uploaded, the dataset becomes available for analysis job configuration.

6.4 Data requirements

Datasets should meet the following criteria:

- Structured tabular format
- Consistent schema across all rows
- Numerical values only (integer or float)
- No missing values in critical variables

6.5 Data size constraints

Maximum dataset size depends on the active subscription plan.

Users should refer to their subscription terms for applicable limits.

6.6 Recommendations

For optimal results:

- Convert categorical or boolean data into numerical representations before upload
- Ensure data consistency and integrity prior to processing
- Remove or anonymize sensitive identifiers when not required for analysis
- Validate datasets locally before upload to reduce processing errors

7 Running an Analysis

7.1 End-to-end workflow

Running an analysis follows a structured, step-by-step process combining dataset validation, configuration, and execution.

The complete workflow is:

1. Upload a Parquet dataset
2. Backend validation of dataset format and structure
3. Presentation of available configuration options and associated pricing
4. User confirmation of execution (role ≥ Executor required)
5. Job submission and execution

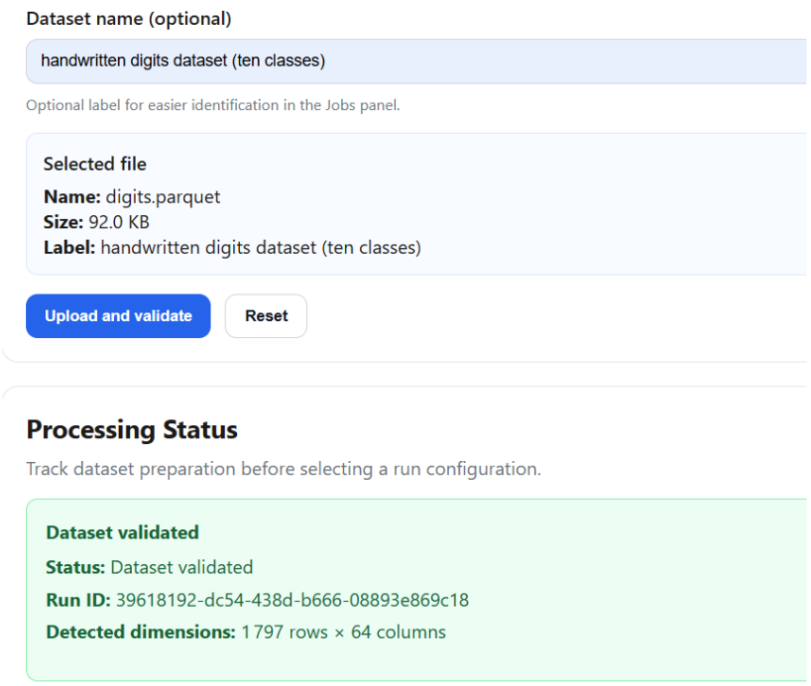
7.2 Dataset validation

Once a dataset is uploaded, it is automatically validated by the platform.

Validation checks include:

- file format compliance (Parquet)
- data type verification (integer / float only)
- structural integrity of the dataset

If validation fails, the dataset cannot be used for analysis.



7.3 Configuration and pricing

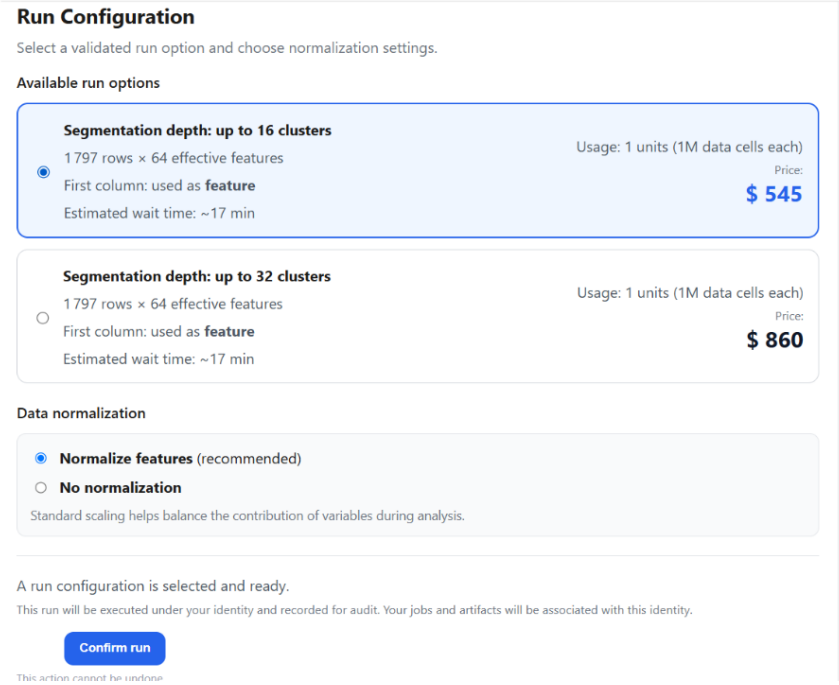
After successful validation, the platform presents the available execution options.

These include:

- **Normalization option**
  Enable or disable normalization prior to processing.
- **Cluster range (K)**
  Available ranges:
  - K = 1 to 16
  - K = 1 to 32

For the selected dataset, the platform also displays the **associated execution cost**.

This allows users to review configuration choices and pricing **before confirming the analysis**.



7.4 Execution authorization

To launch an analysis:

- the user must have a role **greater than or equal to Executor**
- the user must explicitly confirm the execution

This step ensures that all analysis runs are intentional and authorized.

7.5 Job execution

Once confirmed, the job is submitted and processed by the platform.

7.6 Execution lifecycle

A job transitions through the following states:

- **Pending**
  The job has been submitted and is waiting for execution.
- **Running**
  The analysis is currently being executed.
- **Completed**
  The job has successfully finished and results are available.
- **Failed**
  The job execution did not complete successfully.

7.7 Execution model

Each job represents a complete and independent execution.

- Parameters are fixed at confirmation time
- Execution is deterministic for identical inputs
- Results are reproducible across runs

This model ensures consistency, traceability, and controlled usage.

8 Results and Artifacts

8.1 Overview

MathIAs+ Structural Intelligence Studio transforms structured datasets into deterministic, auditable analytical outputs designed to support human-supervised decision-making.

The platform provides:

- deterministic and reproducible artifacts derived from structured data processing
- multi-K clustering evaluation supporting informed selection of cluster configurations
- decision-support outputs aligned with business, expert, and governance requirements
- artifacts structured for downstream analytics, validation, and integration workflows

All outputs are generated within a controlled execution context and remain fully traceable to the originating job.

8.2 Generated artifacts

Each completed job produces a set of complementary artifacts.

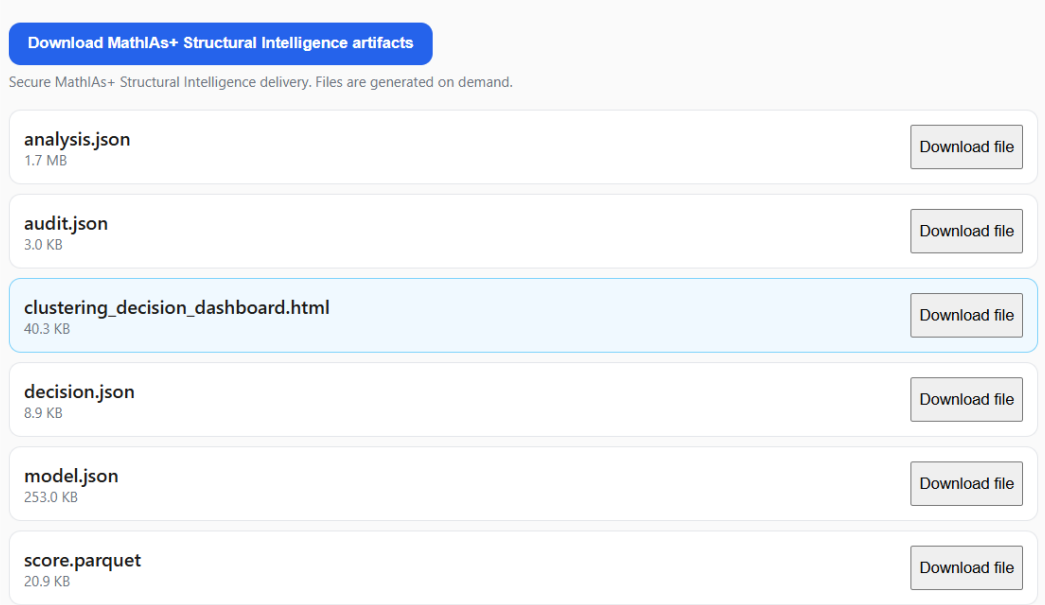
These artifacts are designed to support:

- decision-making
- expert analysis
- governance and audit requirements

All artifacts are associated with:

- a unique **Job ID**
- the execution parameters
- the processed dataset

A cryptographic hash of the input dataset is recorded to ensure data integrity and reproducibility.



8.3 Artifact list and description

The platform generates the following artifacts:

- **clustering_decision_dashboard.html**
  Interactive dashboard designed for business users and decision-makers.
  It enables exploration of clustering results and supports human-supervised selection of the number of clusters (K).

- **decision.json**
  Structured decision table used by the dashboard.
  It provides a traceable representation of clustering evaluation across K values.

- **analysis.json**

Comprehensive analytical output intended for expert users.

This file contains detailed structural evaluation results and may be large in size.

A formal JSON schema is provided to support interpretation and integration:
👉 View analysis schema

The schema defines the structure and semantics of the generated analysis data, enabling:

- programmatic access to results
- integration into downstream analytical pipelines
- validation and consistency checks

- **audit.json**
  Execution trace artifact ensuring auditability.
  It includes:
    - Job ID
    - dataset hash
    - execution parameters
    - configuration details

This artifact enables full traceability and reproducibility.

- **model.json**
  Deterministic clustering model containing centroids and scaling parameters.
  This model allows reproducible assignment of new data using a standard distance-based approach aligned with clustering assignment principles.

- **score.parquet**
  Dataset-level output containing cluster assignments for each observation across evaluated K values.
    - preserves the original index column if provided
    - enables downstream analysis and integration

8.4 Artifact access

Artifacts are:

- stored securely
- linked to the originating job
- available for download via the Results section

Access to artifacts is governed by the user role and subscription policy.

8.5 Traceability

Each artifact is systematically linked to its execution context.

This ensures:

- **Reproducibility**
  Identical datasets and configurations produce identical outputs.

- **Auditability**
  All processing steps and parameters are recorded.
- **Traceable decision support**
  Decisions derived from the analysis can be traced back to their underlying data and execution context.

8.6 Documentation and example dataset

The documentation provides a complete example using a reference dataset.

This example demonstrates:

- the full analysis workflow
- the generated artifacts
- the structure and interpretation of outputs

To support reproducibility and user understanding, a **downloadable package (ZIP)** is made available:

👉 **Download example artifacts**

This package includes:

- the complete set of generated artifacts
- example outputs corresponding to a full analysis run

This enables users to:

- explore the structure of generated artifacts
- reproduce the analytical workflow in a controlled environment
- understand the relationship between dataset, configuration, and outputs

The example is designed to provide a **reference implementation** of the platform's analytical process.

9 Dashboard and Interpretation

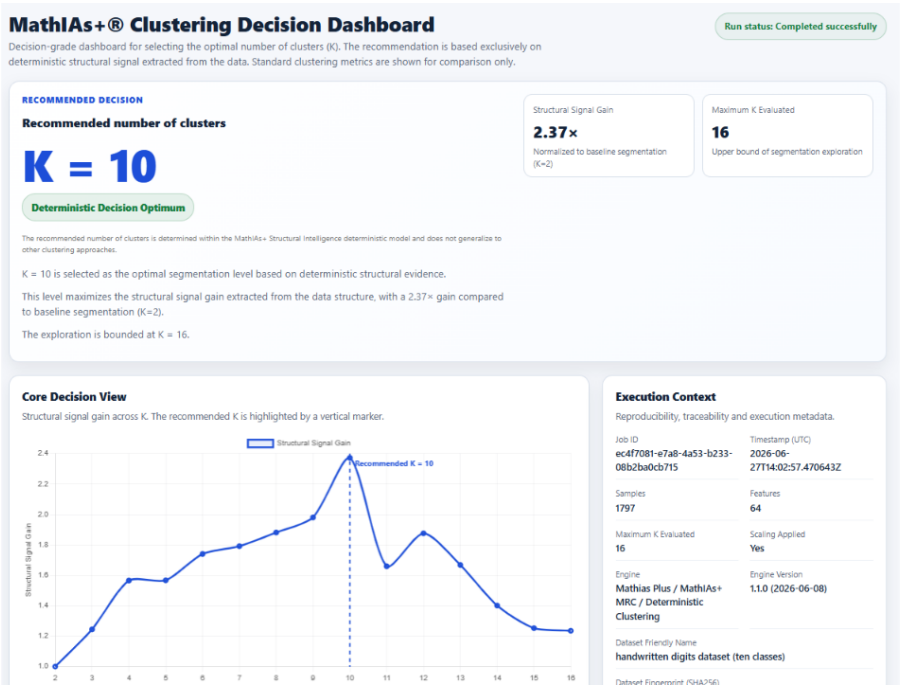
9.1 Dashboard purpose

The platform provides an interactive dashboard in HTML format (**clustering_decision_dashboard.html**) designed for business users and decision-makers.

This dashboard presents a high-level synthesis of the analysis results and supports **human-supervised selection of the number of clusters (K)**.

It enables users to:

- visualize clustering evaluation across multiple K values
- interpret structural segmentation results
- support decision-making based on consistent and traceable outputs



9.2 Multi-K analysis and recommendation

The dashboard presents a comparative analysis of clustering quality across the evaluated range of K values.

This analysis is based on the structured data contained in the **decision.json** artifact.

The dashboard:

- displays aggregated evaluation scores for each K
- highlights structural patterns across cluster configurations
- provides guidance for selecting an appropriate number of clusters

Based on this information, the dashboard identifies a **“K recommended”** value.

This recommendation:

- is derived from deterministic evaluation results
- reflects the underlying structural analysis
- remains fully traceable through the decision.json artifact

9.3 Reading results

Users can interact with the dashboard to analyze:

- **Clustering quality across K values**
  Comparative view of partition quality for different numbers of clusters.
- **Structural segmentation**
  Identification of stable groupings within the dataset.
- **Evaluation metrics**
  Aggregated measures supporting interpretation of clustering results.
- **Decision guidance**
  Visual and structured support for selecting an appropriate K value.

9.4 Relationship with artifacts

The dashboard is directly connected to the generated artifacts:

- **decision.json**
  Provides the structured data used to compute and display clustering evaluation and the recommended K.
- **audit.json**
  Ensures traceability of the decision process and execution context.

The dashboard acts as a **readable and interactive layer** built on top of these artifacts.

9.5 Interpretation model

The platform does not automatically enforce a final decision.

Instead:

- the system provides **deterministic evaluation outputs**
- the dashboard presents a **recommended K**
- the final decision remains under **user supervision**

This model ensures:

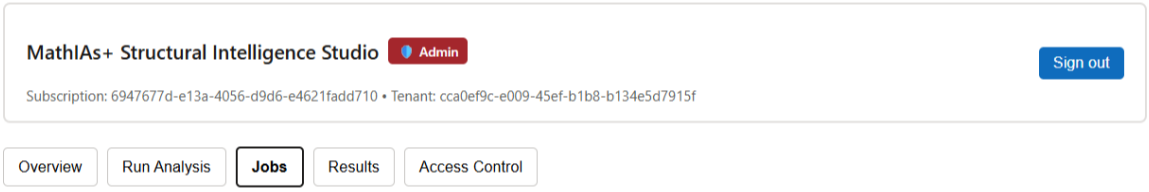
- transparency of analytical results

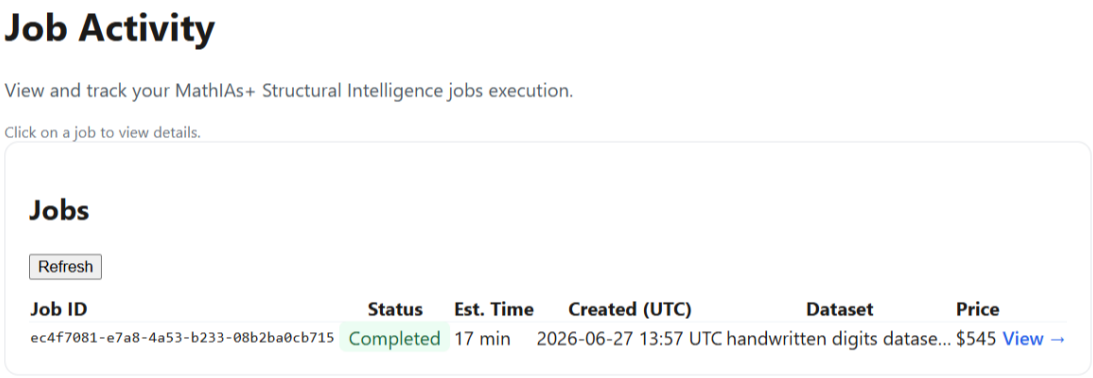
- consistency across executions
- alignment with enterprise governance and decision processes

10 Job Monitoring

The Job Activity interface allows users to:

- Track execution status
- Access historical jobs
- Review completed analyses





11 Security and Data Handling

11.1 Authentication

Access to the platform is secured through:

- **Microsoft Entra ID (Single Sign-On)**

Authentication is fully delegated to the organization's identity provider, ensuring secure and controlled access.

11.2 Data storage

The platform ensures secure handling of datasets and generated outputs.

- **Secure storage**
  Uploaded datasets and generated artifacts are stored using secure storage mechanisms.
- **Controlled access**
  Access to datasets and artifacts is governed by role-based access control (RBAC) and subscription-level policies.

- **Artifact integrity and traceability**
  Generated artifacts remain associated with their originating job and execution context.

A **cryptographic hash of the input dataset** is recorded and included in the generated artifacts to ensure data integrity and reproducibility.

11.3 Input data retention policy

Uploaded input datasets (Parquet files) are **automatically deleted** after processing.

- Input datasets are retained for a **maximum of 4 days after upload**
- After this period, datasets are permanently removed from the platform

This policy ensures:

- minimization of data retention
- reduced exposure of sensitive information
- alignment with data protection best practices

11.4 Data residency

The platform supports regional data hosting.

Available deployment regions include:

- **European Union**
- **United States**

Data residency is determined by the selected subscription plan.

All data processing and storage operations are performed within the region associated with the active plan.

11.5 Security and governance principles

The platform is designed to support enterprise security and compliance requirements:

- access control enforced via Microsoft Entra ID and RBAC
- traceability of all executions and data transformations
- controlled lifecycle of datasets and artifacts
- integrity verification through dataset hashing
- separation between input data, processing, and generated output

12 Best Practices

To ensure optimal usage and reliable analytical outcomes:

- **Use structured and validated datasets**
  Ensure datasets comply with platform requirements (Parquet format, numerical types only) and are validated prior to upload.
- **Use clear dataset identification**
  Assign a meaningful dataset name and use the dataset hash (available in generated artifacts) to ensure traceability across analyses and internal workflows.
- **Verify data before upload**
  Check data consistency, completeness, and numerical integrity to avoid execution issues and unreliable results.
- **Execute analyses on controlled datasets**
  Use well-defined and governed datasets, especially in production or decision-critical environments.
- **Manage normalization appropriately**
  If data has already been pre-scaled or normalized externally, disable normalization during configuration to avoid unintended transformations.
- **Prepare data for weighting or scaling strategies**
  When specific feature weighting is required, prepare the dataset accordingly before upload (e.g. apply pre-scaling or transformations upstream).
- **Retain artifacts for audit and traceability**
  Keep generated artifacts (including audit.json and dataset hash references) to support reproducibility, validation, and compliance requirements.

13 Limitations and Scope

The platform operates under the following conditions:

- Requires structured datasets compliant with platform specifications
- Output quality depends directly on the quality, consistency, and relevance of input data

- **Decision support scope**

MathIAs+ Structural Intelligence Studio is designed as a **decision-support tool**.

- The platform provides deterministic analytical outputs and structured evaluation results
- These outputs are intended to support **expert analysis and human-supervised decision-making**

The platform:

- **does not automate final decisions**
- **does not replace human expertise or judgment**

- **No outcome guarantee**

The service provides analytical outputs based on input data and configured parameters.

It:

- does not guarantee specific business outcomes
- does not ensure performance improvements or operational results

All decisions based on the platform outputs remain under the responsibility of the user.

14 Support and Contact

For support or inquiries:

👉 https://www.mathias-plus.fr/contact-form.html

Support is provided by Mathias Plus SAS.

15 Publisher

MathIAs+ Structural Intelligence Studio is developed and operated by:

Mathias Plus SAS
Limoges, France

The company is responsible for the development, operation, and support of the platform.

For inquiries, support, or access requests:
👉 Contact form

- **Document information**
- Product: MathIAs+ Structural Intelligence Studio
- Document type: User Documentation
- Version: 1.1
- Date: June 2026

Legal notice

This document is provided for informational purposes only. It does not constitute a contractual commitment and does not override the terms of the applicable subscription agreement and Azure Marketplace offer.

16 Annex – Expert Description of analysis.json Schema

The **analysis.json** file is the structured output of a clustering analysis performed by MathIAs+ Structural Intelligence Studio.

It provides a **deterministic, multi-resolution and structurally validated representation** of the clustering process across several values of K.

A detailed schema definition is available via the referenced JSON schema.

👉 View analysis schema

16.1 Conceptual Overview

The schema is designed around three core principles:

- **Multi-K analysis**
  Each clustering solution is evaluated independently for different values of K.
- **Cumulative structural interpretation**
  Results are interpreted not only per K, but across all partitions for K ≤ current K.
- **Deterministic auditability**
  All indicators are designed to be reproducible, explainable, and auditable.

16.2 Multi-Resolution Structure

The main container analysis_by_K organizes results by number of clusters:

- Each key corresponds to a specific value of K
- Each K block contains a **complete analytical state** of the clustering

This design enables:

- comparison across segmentation levels
- identification of stable structural patterns
- progressive refinement analysis

16.3 Analytical Layers

Each K-level combines three complementary layers of analysis:

16.3.1 Statistical Layer

Standard clustering metrics provide a baseline evaluation of:

- intra-cluster compactness
- inter-cluster separation
- global variance distribution

These indicators ensure compatibility with classical clustering evaluation frameworks.

16.3.2 Structural Layer

The schema introduces a structural interpretation of clustering based on relationships between clusters:

- **Pairwise structural scores (MPS)**
  Proprietary scores defined in the range [0,1], evaluating the consistency between cluster pairs
- **Structural signal**
  Aggregated indicators identifying:
  - global separability strength
  - worst-case structural risk
  - critical cluster pairs
- **Connected components**
  Graph-based representation of cluster interactions, highlighting:
  - independent substructures
  - zones of structural ambiguity

This layer captures properties that are not accessible through distance-based metrics alone.

- **Connected components**

  Graph-based representation of cluster interactions, highlighting:

  - independent substructures
  - zones of structural ambiguity

  Clusters are represented as nodes, with edges reflecting structural interactions derived from pairwise analysis.

Each connected component corresponds to a **set of clusters that are structurally linked**, meaning that their separation cannot be considered fully independent. Conversely, distinct components indicate groups of clusters that are structurally disconnected and behave as independent subspaces.

This representation enables:

- identification of **cluster families sharing structural dependencies**
- detection of **weak separation regimes**, where multiple clusters remain interconnected
- distinction between **globally well-separated structures** and **locally entangled regions**

This layer captures properties that are not accessible through distance-based metrics alone, in particular when separability cannot be fully characterized by convexity assumptions.

16.3.3 Explainability Layer

The schema integrates interpretability mechanisms at both global and local level:

- **CART surrogate model**
  Decision tree approximating cluster assignments
  → provides interpretable decision rules and feature importance
- **Cluster statistics**
  Statistical characterization of each cluster:
  - discriminant features
  - statistical significance
  - direction of deviation from global distribution
- **Explainability indicators**
  Quantification of how well clusters can be described using the available features

This layer ensures that clustering results can be **understood and justified in operational contexts**.

16.4 Deterministic Sanity Check

Each clustering solution is benchmarked against a controlled reference:

- k-means++ initialization
- n_init = 1
- fixed random seed

An alert is triggered when the relative deviation in inertia exceeds a defined threshold (2%).

This mechanism ensures:

- detection of unstable solutions
- reproducibility validation
- consistency control across executions

16.5 Multi-K Structural Indicator (Atomicity)

Atomicity is defined as:

- the number of distinct classes observed across all partitions for K ≤ current K

It is a **cumulative indicator**, not a per-K metric.

Atomicity measures:

- segmentation granularity
- structural refinement trajectory
- consistency of cluster emergence across resolutions

16.6 Positioning of the Schema

The analysis.json schema is not limited to a clustering output.
It implements a **structured analytical framework combining**:

- statistical validation
- structural consistency analysis
- explainability and auditability

This enables:

- deterministic interpretation of clustering results
- identification of structurally robust segmentations
- production of decision-grade analytical outputs

16.7 Key Characteristics

- **Deterministic**: reproducible outputs with explicit sanity checks
- **Multi-resolution**: evaluation across multiple values of K
- **Structural**: analysis based on inter-cluster relationships
- **Explainable**: integration of interpretability mechanisms
- **Audit-ready**: explicit indicators and validation rules