## 

**Table of contents**

**-------------------------------------------------------------------------------------------------------------**

[1. Introduction](#_Toc623737254)

[1.1. Overview & Value Proposition](#_Toc566140456)

[1.2. Target Audience](#_Toc1101232888)

[1.3. System Architecture](#_Toc511959407)

[2. Installation & Setup](#_Toc780215413)

[2.1. Running the Hosted Demo](#_Toc996981195)

[2.2. Local Installation](#_Toc1269200380)

[2.3. Security & Data Privacy Note](#_Toc1776371775)

[3. Core Workflow: Detailed Instructions](#_Toc1199629065)

[3.1. Phase 1: Uploading and Processing Data](#_Toc873554677)

[3.2. Phase 2: Configuring Analysis Parameters](#_Toc207457269)

[3.3. Phase 3: Running Analysis](#_Toc1507439927)

[3.3.1. Attribute Analysis & OAM Creation](#_Toc540853190)

[3.3.2. Student Ranking](#_Toc1072091500)

[3.3.3. COCO Y0 Analysis & Validation](#_Toc1194943358)

[3.4. Phase 4: Visualizing Results](#_Toc1843748185)

[3.4.1. Attribute Distribution Analysis](#_Toc1884362260)

[3.4.2. Student Performance Comparison](#_Toc2009312498)

[3.4.3. Above vs Below Cohort Average](#_Toc1738802625)

[3.4.4. Top Performers by Attribute](#_Toc149086245)

[3.4.5. Category-wise Attribute Analysis](#_Toc1057369782)

[3.4.6. Exam Attribute Focus](#_Toc1480457734)

[3.4.7. Ranking Insights](#_Toc1800954394)

[3.4.8. COCO & Validation](#_Toc1615368795)

[3.5. Phase 5: Generating AI Insights & Watchlists](#_Toc867173491)

[3.5.1. Using the AI Insights Interface](#_Toc1938686618)

[4. Exports, Session History & Collaboration](#_Toc269452053)

[5. Troubleshooting & Quality Gates](#_Toc1485888373)

[6. Appendices](#_Toc2029496831)

[6.1. Appendix A – Attribute Dictionary](#_Toc1608472351)

[6.2. Appendix B – Configuration Reference](#_Toc2040412553)

[6.3. Appendix C – Glossary & Keyword Coverage](#_Toc767712060)

# **1. Introduction**

## **1.1. Overview & Value Proposition**

The Moodle Log Analyzer is a comprehensive analysis pipeline that transforms raw Moodle discussion exports into actionable, instructor-ready insights. Built as a Streamlit application, it automates the evaluation of student participation by converting qualitative discussion data into quantitative metrics, validated rankings, and AI-powered narratives.

**Core Value:**

* **Objective & Reproducible:** Ensures consistent, transparent evaluation by providing a deterministic pipeline from raw data to final reports.
* **Actionable Insights:** Surfaces both quantitative metrics (e.g., engagement scores, deadline misses) and qualitative narratives (e.g., AI-generated summaries and watchlists) to support intervention decisions.
* **Self-Contained & Secure:** Requires no external database; all processing occurs locally on your machine. Data privacy is maintained, with exports triggered only by user action.
* **Configurable & Reusable:** Save analysis configurations as JSON files, export all intermediate artifacts (CSVs, tables), and share reproducible scenarios with colleagues.

## **1.2. Target Audience**

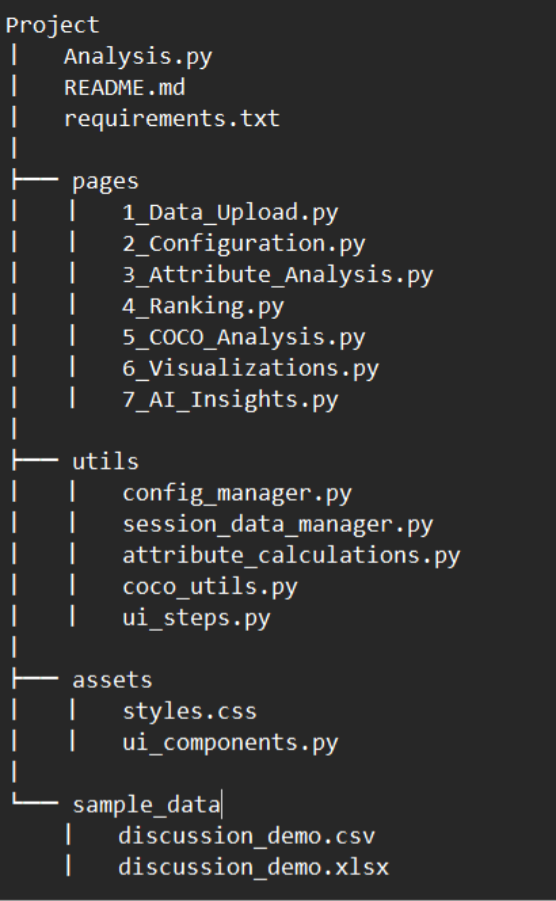
* **Course Instructors & Professors:** Manage and evaluate participation in Moodle forums, identify at-risk students, and gather evidence for grading.
* **Educational Researchers:** Analyse engagement metrics systematically for publications, studies, or internal program assessment.
* **Thesis Supervisors & Examiners:** Validate that specific discussion criteria or "use-case keywords" are met with auditable evidence from the analysis pipeline.

## **1.3. System Architecture**

The application is a modular **Streamlit** web application. Its architecture ensures separation of concerns, efficient data flow, and maintainability.

**Key Components** (FIGURE 1)**:**

* **Modular Interface:** The user workflow is divided into logical pages (e.g., Upload, Configuration, Analysis, Visualization) located in the /pages directory.
* **Utility Modules:** Reusable functions for data processing, calculations, and API calls are organized under /utils.
* **State Management:** The SessionDataManager class caches all user data, computed attributes, rankings, and results within the user's session.
* **Centralized Configuration:** The ConfigManager class handles all settings—professor lists, exam deadlines, analysis thresholds, and AI rules—providing a single source.

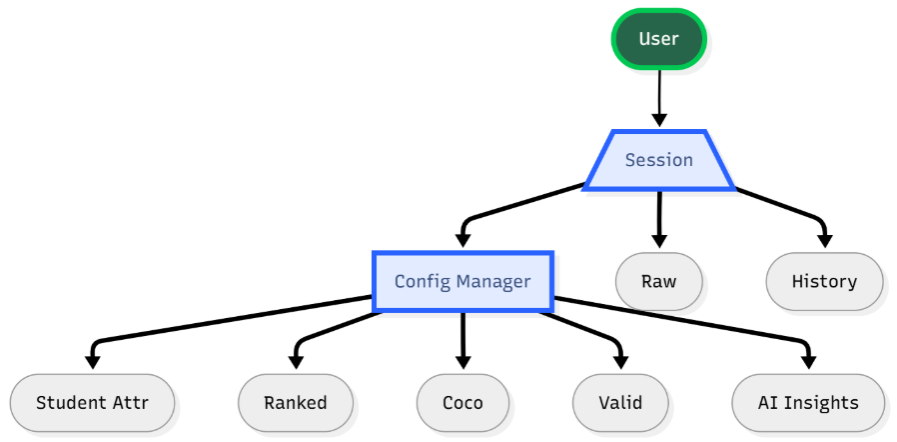


*FIGURE 1. App structure*

Following flow-chart (Figure 2) shows the flow of the pages.

*FIGURE 2. Flow of the pages*

Following flow-chart (Figure 3) shows how the session state is structured and managed.

*FIGURE 3. Session state structure*

# **2. Installation & Setup**

You can access the Moodle Log Analyzer through a hosted demo or by running it locally on your machine. Local installation provides full control over data privacy and allows for offline operation.

2.1. Running the Hosted Demo

For a quick evaluation without installation, access the public demo:

URL: <https://moodleloganalysisdemo-2025.streamlit.app/>

Note: The demo uses a shared server with limited resources. For production analysis with full performance and data privacy, a local installation is recommended.

2.2. Local Installation

For full functionality, data privacy, and optimal performance, install and run the application locally.

**Step 1: Get the Source Code**

Clone the repository to your local machine:

**git clone** [**https://github.com/Shagai-hub/Moodle\_Log\_Analysis\_DEMO.git**](https://github.com/Shagai-hub/Moodle_Log_Analysis_DEMO.git)**cd Moodle\_Log\_Analysis\_DEMO**

**Step 2: Verify Prerequisites**

Ensure your system meets these requirements before proceeding:

Software:

* Python 3.10 or higher
* pip package manager
* A modern web browser (Chrome, Firefox, Safari, Edge)

Hardware (Minimum):

* 8 GB RAM
* Multi-core CPU
* Hardware (Recommended for ML Features):
* 16+ GB RAM
* GPU (significantly accelerates transformer-based attribute computation)

Internet Access: Required only for the initial download of optional Hugging Face models and for COCO Y0 web requests. All core computations function offline.

**Step 3: Install Dependencies**

From the application's root directory, install all required Python libraries:

**pip install -r requirements.txt**

This command installs the core stack, including pandas, plotly, streamlit, transformers, and requests.

**Step 4: Launch the Application**

Start the Streamlit server:

**streamlit run Analysis.py**

The application will launch and automatically open in your default web browser. If it doesn't, navigate to the local URL displayed in your terminal (typically http://localhost:8501).

2.3. Security & Data Privacy Note

**All processing occurs locally.** Your Moodle data never leaves your machine unless you explicitly trigger an optional web request (for COCO analysis or model download). You maintain full control over all exported files.

# **3. Core Workflow: Detailed Instructions**

This section provides comprehensive, step-by-step guidance for the entire analytical pipeline.

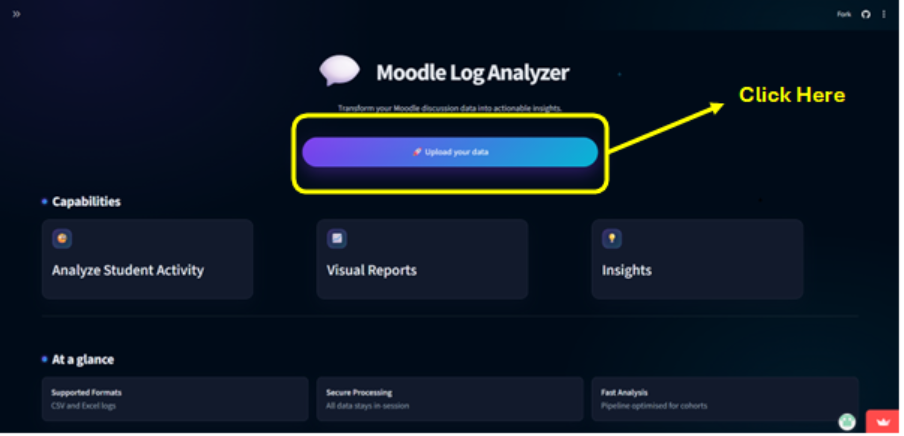
**Example scenario:** Two professors (*Professor\_1* and *Professor\_2*) need to evaluate forum participation at the end of a semester. They will:

1. Import the latest Moodle discussion export.
2. Tag their own accounts to exclude them from analysis.
3. Set key exam deadlines.
4. Compute comprehensive student metrics.
5. Rank students using configurable fairness rules.
6. Run COCO Y0 validation for algorithmic fairness checking.
7. Explore interactive dashboards.
8. Generate AI-driven watchlists for targeted interventions.

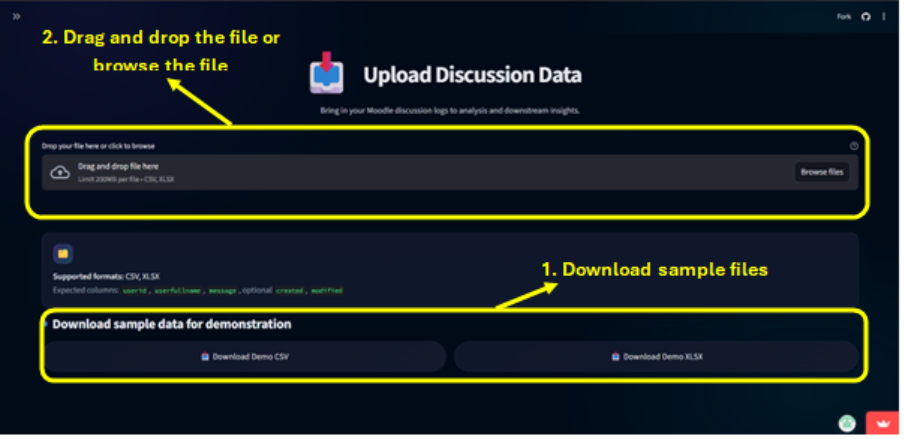
**Inputs:** Moodle CSV or XLSX export file.  
**Outputs:** Processed CSV, Object Attribute Matrix (OAM), ranking tables, COCO validation results, interactive dashboards, AI summary report.

3.1. Phase 1: Uploading and Processing Data

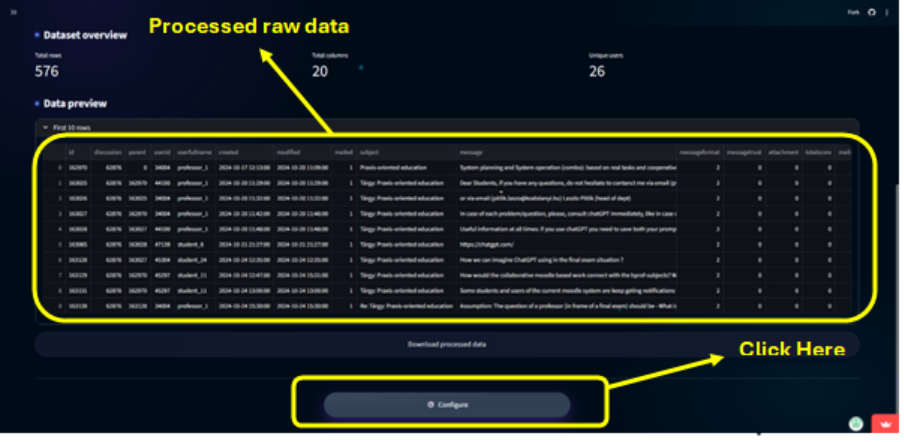
1. **Navigate to the Upload Page:** From the **Landing Page** (Figure 4), click the **Upload your data** button.

*Figure 4: The Landing Page provides an overview of application capabilities and the entry point to the data upload workflow.*

1. **Upload Your File:** On the **Data Upload** page (Figure 5), drag and drop your Moodle export file (CSV or XLSX) or click to browse. The system automatically validates the presence of mandatory columns (e.g., userid, message, created).

*Figure 5: The Data Upload page interface showing the file upload area and sample data download options.*

1. **Review Processed Data:** After upload, the system normalizes headers and processes the data. Review the **Data Preview** (Figure 6) to ensure your data was interpreted correctly.

*Figure 6: A preview table showing the first 10 rows of the processed, analysis-ready dataset.*

**Tip:** Sample datasets in various formats are available to download at the bottom of the page for testing.

1. **Proceed:** Click the **Proceed to Configuration** button to continue.

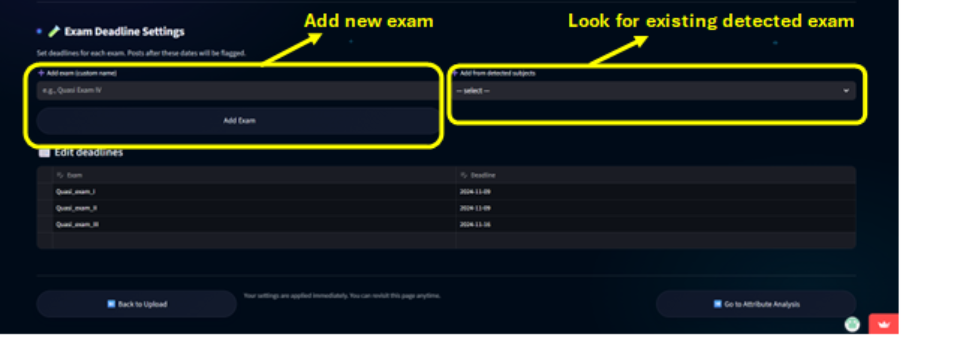
3.2. Phase 2: Configuring Analysis Parameters

Navigate to the **Configuration** page (Figure 7) to define how your data will be analysed. This page contains three tabs: **Professors & Exams**, **Analysis Settings**, and **Export/Import**.

*Figure 7: The main Configuration page with its three primary tabs.*

**Professors & Exams Tab**

* **Professors Sub-tab:** Enter the usernames or full names of instructors (one per line). These accounts will be filtered out from student metrics. The system may suggest names detected with high frequency in your data (Figure 7).
* **Exams Sub-tab:** Define assessment deadlines crucial for calculating "deadline missed" metrics (Figure 8).

*Figure 8: The exam editing portal within the Configuration page, showing a table for adding, editing, and deleting exam deadlines.*

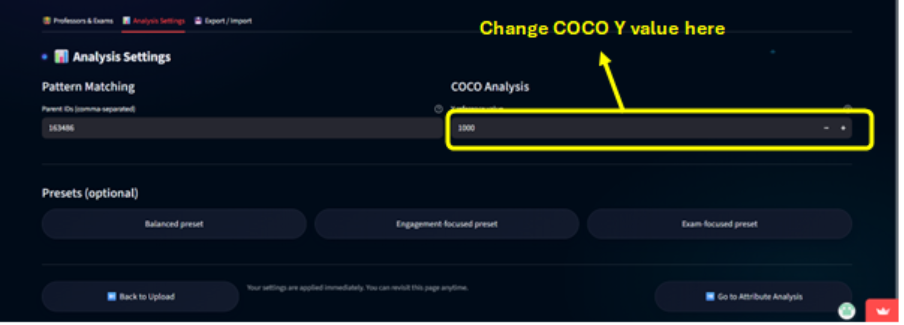
You can add exams manually or select from detected "exam" subjects in your data.

**Analysis Settings Tab**

Configure advanced parameters (Figure 9):

**Pattern IDs:** Enter parent post IDs to compute the Pattern\_followed attribute.

**Y Reference Value:** Set the reference value used for COCO analysis ranking calculations (default: 1000). This can be overridden later on the Ranking page.

*Figure 9: The Analysis Settings tab showing configuration fields for pattern IDs, Y value.*

**Export/Import Tab**

Manage your configuration (Figure 10):

**Export:** Download a JSON snapshot of your current settings for backup or sharing.

**Import:** Upload a previously saved JSON file to restore a configuration.

**Reset:** Return all settings to their default values.

*Figure 10: The Export/Import tab with buttons for downloading, uploading, and resetting the configuration JSON.*

**Note:** For reproducibility in academic assessment, document your professor list and exam deadlines in appendices. Evaluators can cross-reference them with the exported configuration JSON.

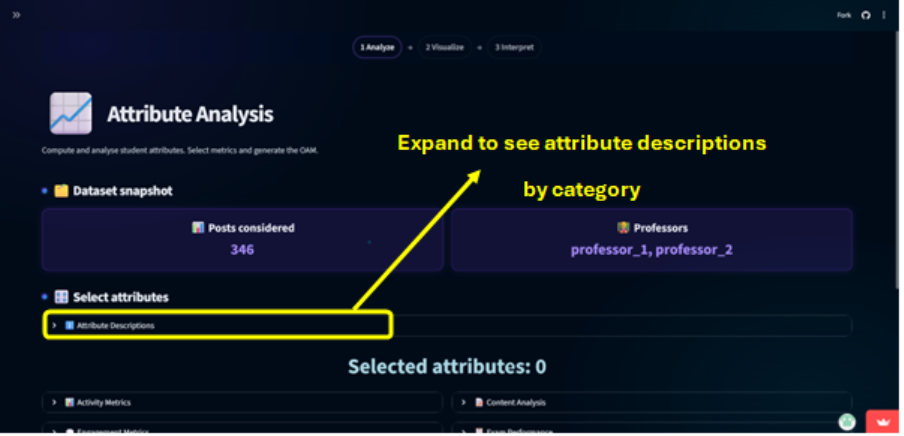
3.3. Phase 3: Running Analysis

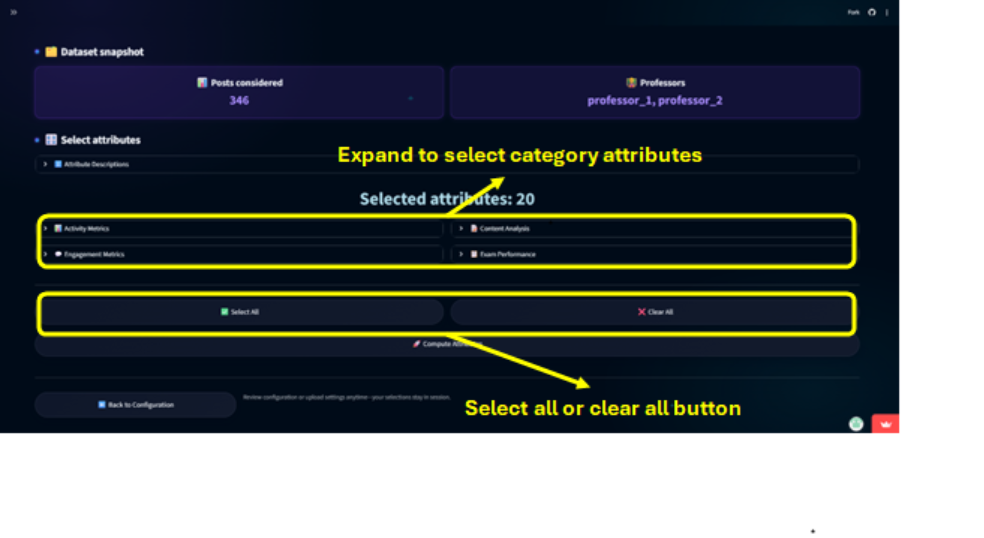
This phase consists of three sequential steps: Attribute Analysis, Student Ranking, and COCO Validation.

3.3.1. Attribute Analysis & OAM Creation

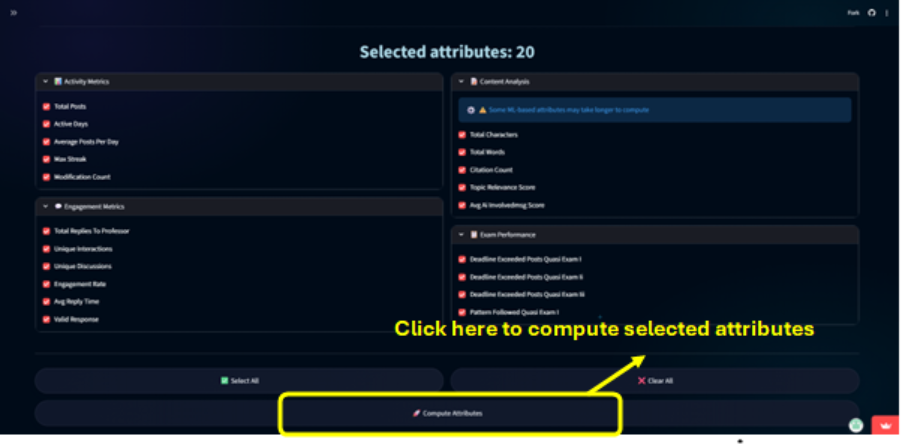
The **Attribute Analysis** page (Figure 11,12) filters out professors and displays available metrics in expandable categories: **Activity**, **Engagement**, **Content**, and **Exam**.

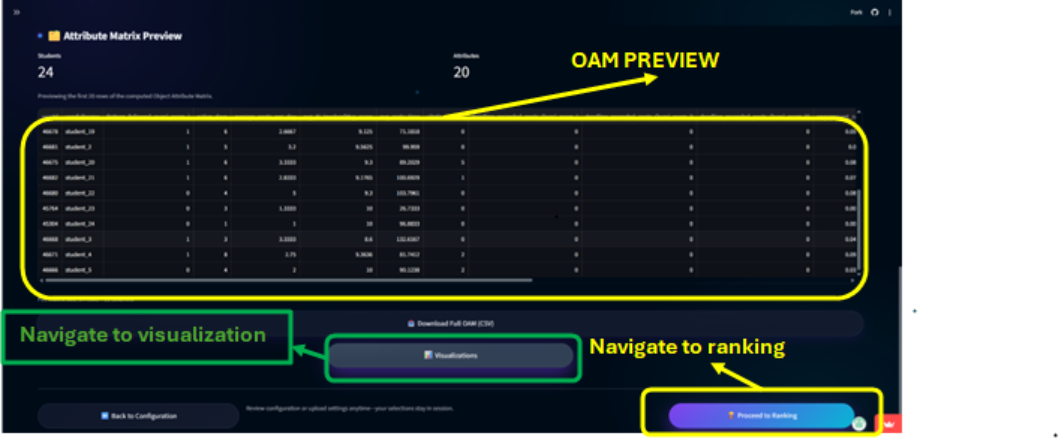
1. **Select Attributes:** Inspect attribute descriptions and select/deselect the metrics you wish to compute. Use the **Select All** or **Clear All** buttons for bulk operations (Figure 11, 12).

*Figure 11: The Attribute Analysis page*

*Figure 12: The attribute selection interface with category headers and bulk action buttons.*

1. **Compute Object Attribute Matrix (OAM):** Click **Compute Attributes**. The system processes your selections, merges result by userid, and generates the OAM (Figure 13, 14).

*Figure 13: The Attribute Analysis page.*

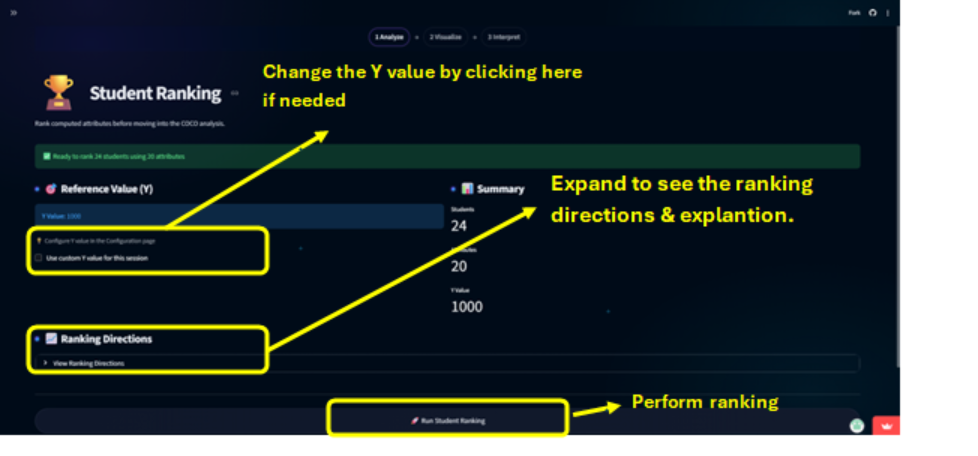
*Figure 14: A preview of the computed Object Attribute Matrix (OAM) table, with key statistics and a button to proceed to Visualization.*

1. **Verify & Proceed:** Review the OAM preview and the displayed student/attribute counts. Click **Proceed to Ranking** to continue or directly navigate to see the visualizations.

3.3.2. Student Ranking

The **Ranking** page (Figure 15) uses the OAM to generate performance rankings.

1. **Review Settings:** The page displays the Y reference value from the configuration. You may enter a temporary override here for this session if needed.
2. **Generate Rankings:** Click **Run Student Ranking**. The system calculates an overall rank and per-attribute ranks for each student (Figure 16).

*Figure 15: The Ranking page with the Y-value input and the main ranking button.*

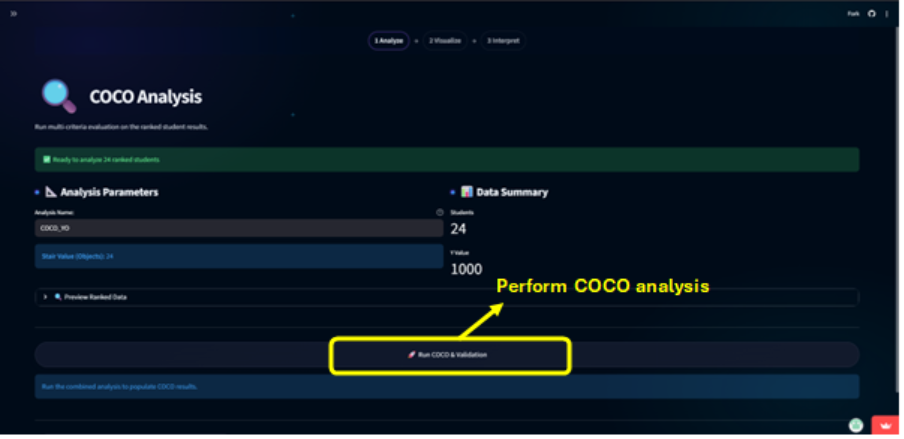
*Figure 16: A preview of the ranked results table, showing student IDs, overall rank, and attribute-wise ranks.*

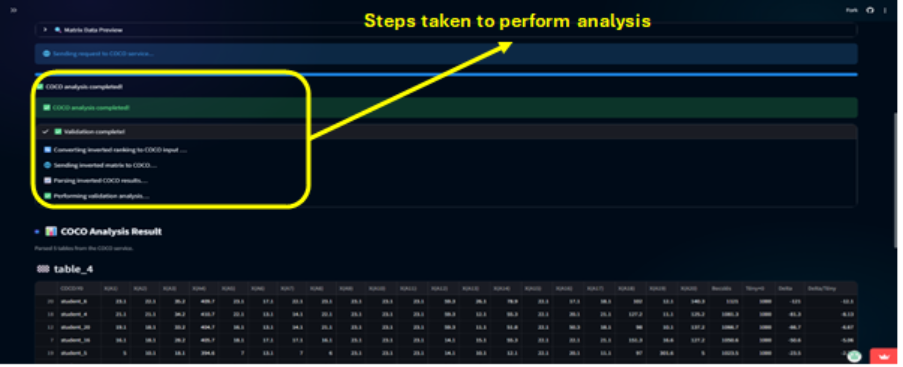
1. **Proceed:** Click **Proceed to COCO Analysis** to analyse the rankings.

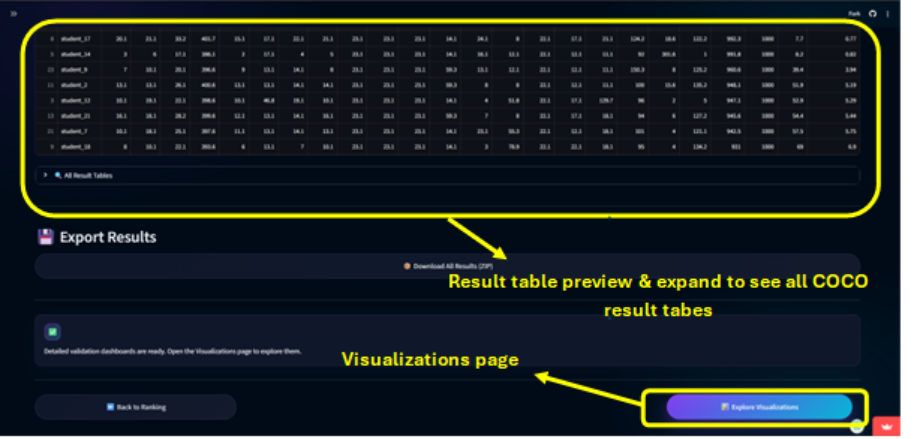
3.3.3. COCO Y0 Analysis & Validation

The **COCO Analysis** page (Figure 17) performs algorithmic fairness validation on your rankings.

* **Configure:** Optionally, enter a custom job name for the session. The page shows the stair value (number of students).
* **Run COCO:** Click **Run COCO Analysis** to submit your ranked data. The system processes and returns COCO /tables (Figure 18, 19).

*Figure 17: The COCO Analysis page with input fields and the main action button.*

*Figure 18: COCO results.*

*Figure 19: The COCO Analysis page results section and a button to navigate to the Visualization page.*

**Complete Analysis:** The analysis phase is now complete.

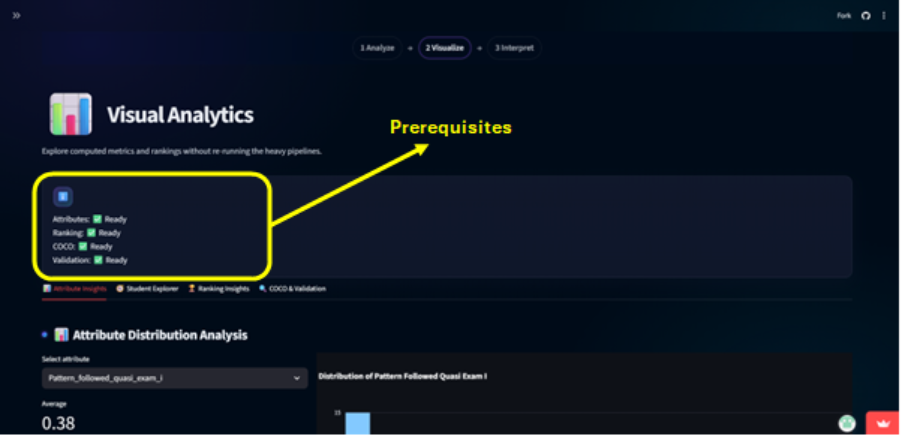
Click the **Visualization** button to explore your results through interactive dashboards (Figure 19).

3.4. Phase 4: Visualizing Results

The **Visualization** page provides interactive visual analytics built from the computed Object Attribute Matrix (OAM), ranking output, COCO results, and validation tables. Visualizations are grouped into tabs and tools that let instructors inspect distributions, compare students, validate COCO outcomes, and export reproducible artifacts. All charts update from cached session data. To refresh figures, re-run the Attribute Analysis, Ranking, or COCO pages.

**Prerequisite:** Attributes must be computed on the **Attribute Analysis** page.

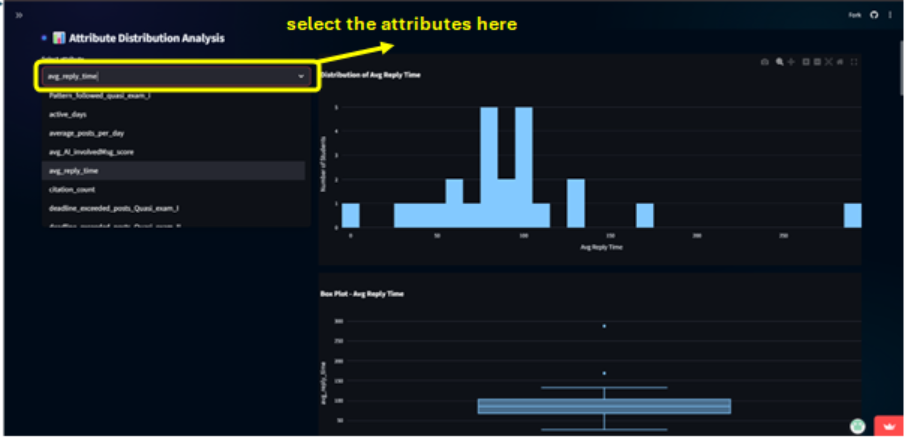
The page displays an info panel showing status for Attributes, Ranking, COCO, and Validation (FIGURE 20).

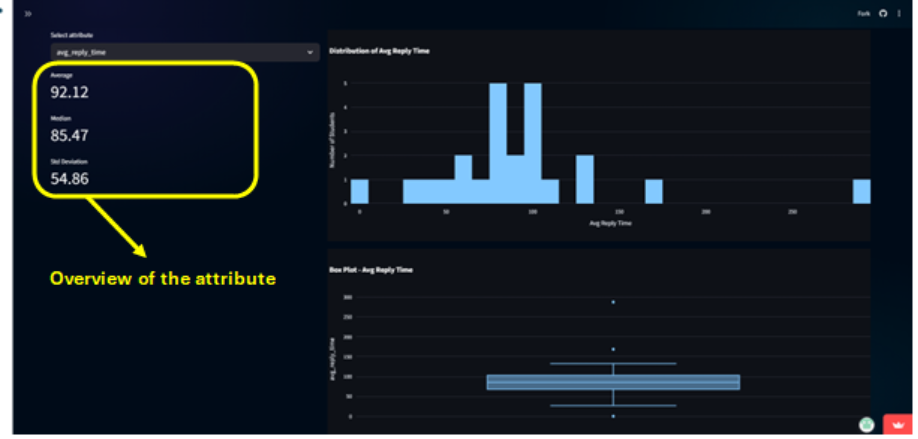
*Figure 20: The Visualizations page landing view with system status panel.*

3.4.1. Attribute Distribution Analysis

Analyse the statistical distribution of any single attribute across the cohort.

* **Charts:** Histogram and Box plot for the chosen attribute (FIGURE 21).
* **Displayed Metrics:** Average, Median, Standard Deviation.
* **What It Shows:** Shape of the distribution (skew, multimodality), spread, and outliers. Provides an overview of the selected attribute (FIGURE 22).
* **How to Interpret:**
  + **Low skew:** Indicates a few students dominate the attribute.
  + **Large IQR or many outliers:** Suggests investigating data quality or specific student edge cases.

*Figure 21: The Attribute Distribution Analysis view with histogram and box plot.*

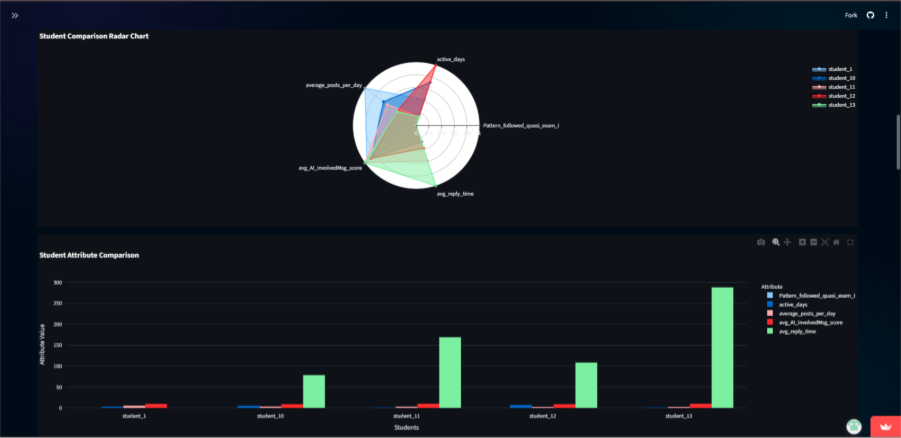
*Figure 22: Overview panel for the selected attribute within the distribution analysis.*

3.4.2. Student Performance Comparison

Compare the relative strengths and weaknesses of 2-5 selected students.

* **Charts:**
  + **Radar chart:** Attributes normalized per attribute by their max value (FIGURE 23).
  + **Grouped bar chart:** Compares absolute attribute values per student (FIGURE 24).
* **What It Shows:** Relative performance profiles across chosen attributes.
* **How to Interpret:** Radar normalization highlights proportional differences; the bar chart provides absolute comparisons.

*Figure 23: Radar chart comparing selected students.*

*Figure 24: Grouped bar chart for absolute value comparison.*

3.4.3. Above vs Below Cohort Average

Get a rapid, color-coded overview of student performance relative to class means.

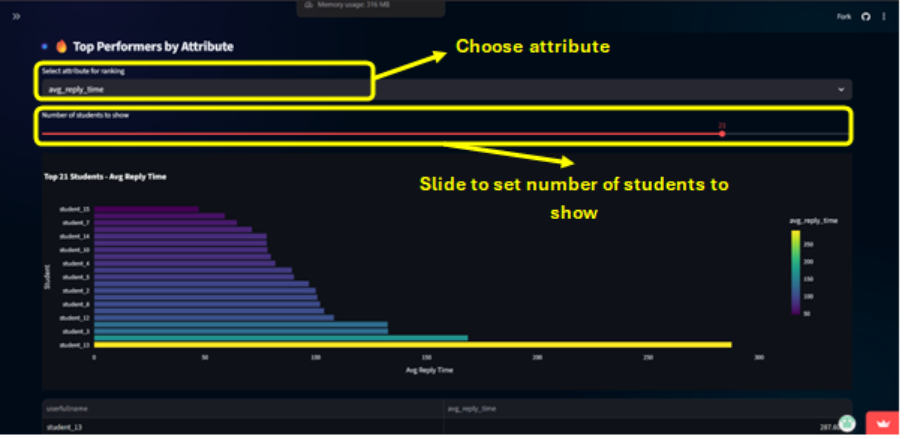
* **Table:** A styled data frame comparing each student's score to the cohort mean for each attribute (FIGURE 25).
* **Visual Encoding:**
  + **Green cell:** Score is **above** cohort average.
  + **Red cell:** Score is **below** cohort average.
* **What It Shows:** Quick visual identification of where students outperform or underperform the cohort mean across all attributes.

*Figure 25: The Above vs Below Cohort Average matrix table.*

3.4.4. Top Performers by Attribute

Create a focused leaderboard for any single metric.

* **Charts:** Horizontal bar chart showing the top \*n\* students for the selected attribute, with a supporting data table below (FIGURE 26).
* **What It Shows:** A clear ranking of students excelling in a specific area.

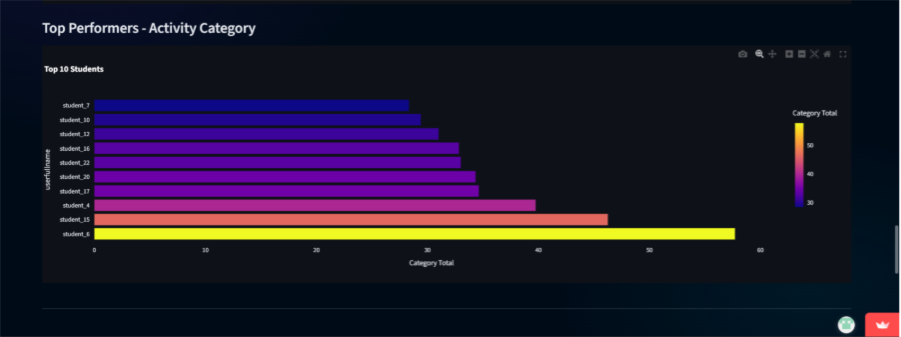
*Figure 26: Top Performers view for a selected attribute.*

3.4.5. Category-wise Attribute Analysis

Evaluate performance across the four logical attribute groupings (Activity, Engagement, Content, Exam).

* **Charts:**
  + Bar chart of average scores for all attributes within a selected category (FIGURE 27).
  + Leaderboard of the top 10 students by their total score within a category (FIGURE 28).
* **What It Shows:** How the cohort performs on grouped dimensions and identifies the leaders in each category.

*Figure 27: Bar chart of category attribute averages.*

*Figure 28: Top 10 students by category total.*

3.4.6. Exam Attribute Focus

Zero in on performance specifically related to exam periods and deadlines.

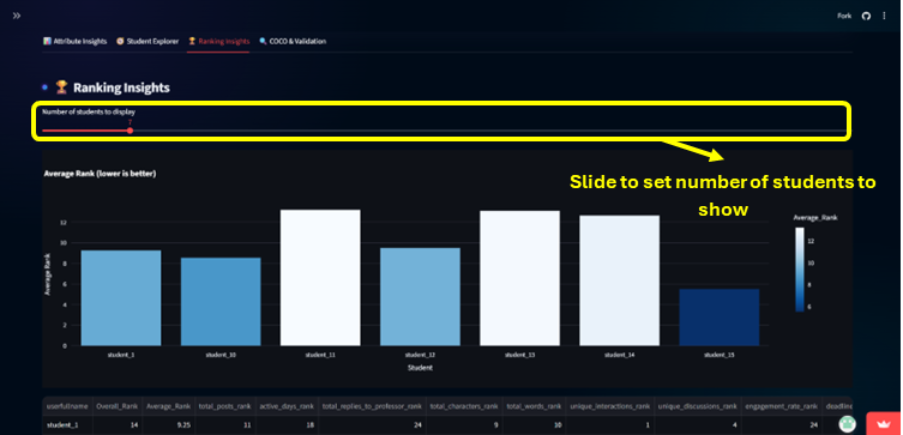
* **Charts:** Two-panel view (FIGURE 29):
  + **Left:** Top 10 students for the selected exam-focused attribute.
  + **Right:** Bottom 10 students requiring attention for the same attribute.
* **What It Shows:** Highlights both exemplary and concerning behavior specific to exam deadlines.

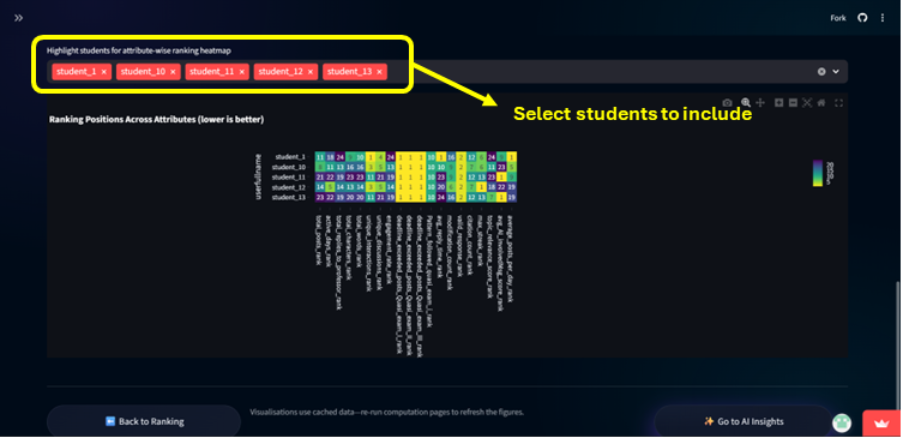
*Figure 29: Exam Attribute Focus dashboard showing top and bottom performers.*

3.4.7. Ranking Insights

Analyse the results of the student ranking process.

* **Precondition:** Requires the ranked result table to be present (from FIGURE 16).
* **Charts and Controls:**
  + **Slider:** To select the number of students to display (FIGURE 30).
  + **Bar Chart:** Visualizes Average Rank (lower is better) for the top students (FIGURE 30).
  + **Data Table:** Shows Overall\_Rank, Average\_Rank, and per-attribute ranks.
  + **Multi-select:** Highlight students to generate an attribute-wise ranking heatmap (FIGURE 31).
* **What It Shows:** Aggregate and per-attribute ranking positions across the cohort. The heatmap exposes attribute-by-attribute rank positions for selected students.
* **How to Interpret:**
  + Low Average\_Rank = better overall performance.
  + Tightly clustered Average\_Rank suggests a need for tiebreakers or additional attributes for better differentiation.

*Figure 30: Ranking Insights dashboard with average rank bar chart and controls.*

Figure 31: Attribute-wise ranking heatmap for selected students.

3.4.8. COCO & Validation

Validate and interpret the results of the COCO Y0 fairness analysis.

**COCO Result Overview**

* **Chart:**
  + **Top COCO Scores:** Horizontal bar chart and table of the top students by Estimations (FIGURE 32).
* **What It Shows:** Identifies the students most strongly recommended by the COCO anti-discrimination algorithm and the distribution of COCO estimates.
* **Actionable:** If COCO finds unexpectedly few students with high scores or too many identical scores, re-check the Y reference value and fairness settings.

*Figure 32: COCO Result Overview dashboard.*

**Validation Results Dashboard**

A comprehensive multi-tab dashboard for validation analysis.

* **Tabs:**
  + **Score Distribution:** Histogram of Estimations colored by validation results (**Valid**/**Invalid**) (Figure 35).
  + **Validation Analysis:** Scatter plot of Original\_Delta vs Inverted\_Delta, colored by validation results with point size mapped to Estimations. Shows delta correlation and status insights (Strong/Moderate/Low consistency) (Figure 33).
  + **Detailed Results:** Table with student names, ranks, scores, deltas, and validation labels.
  + **Review Cases:** Lists invalid cases, counts, percentage, average invalid score, and expandable pattern & delta analysis.
* **What It Shows:** Whether the inversion and COCO checks are consistent and which students failed validation. Original\_Delta vs Inverted\_Delta correlation reveals whether inversion behaved as expected (low correlation typically indicates a valid inversion).

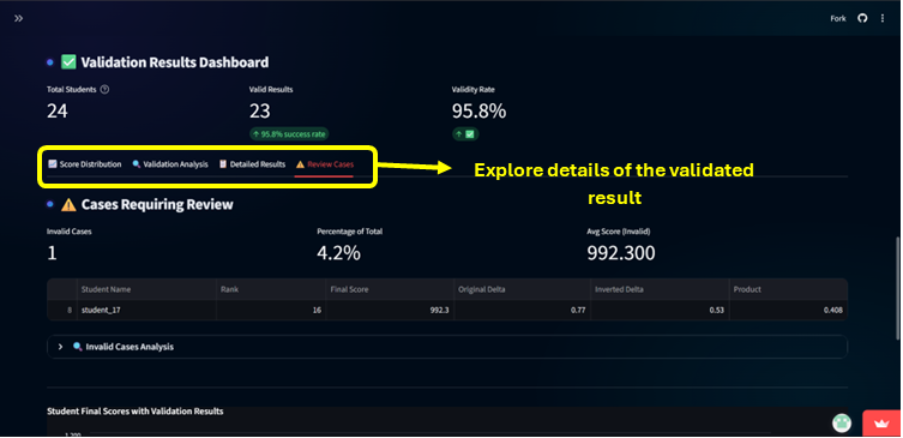
Figure 33: Validation Results Dashboard - Review Cases tab

Figure 34: Validation Results Dashboard - Score Distribution histogram.

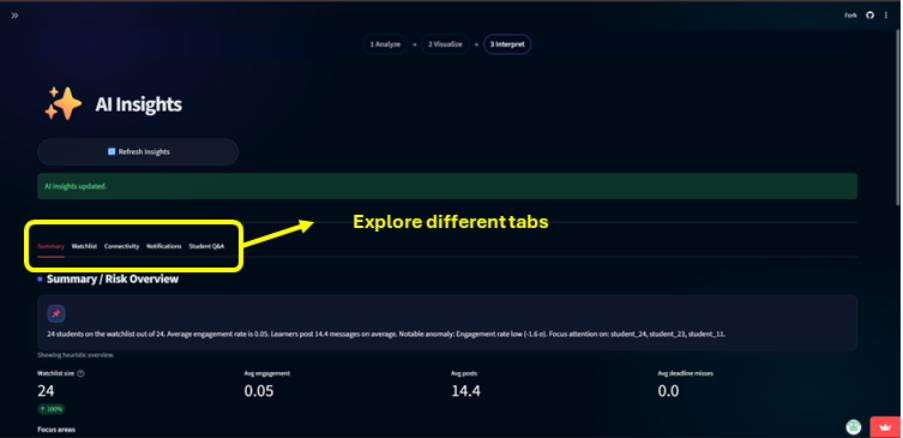
*Figure 35: Final validation bar chart.*

3.5. Phase 5: Generating AI Insights & Watchlists

The **AI Insights** page combines cohort-level data, detects statistical outliers, and builds a prioritized watchlist to guide instructor intervention. It combines heuristic rules, anomaly detection, and generative AI to create actionable narratives and sample communications. All outputs are generated from cached session data.

**Prerequisite:** The Object Attribute Matrix (OAM) must exist (computed on the **Attribute Analysis** page).

**Refresh Data:** Use the **Refresh** button to regenerate all insights if you change attributes, rules, or configuration.

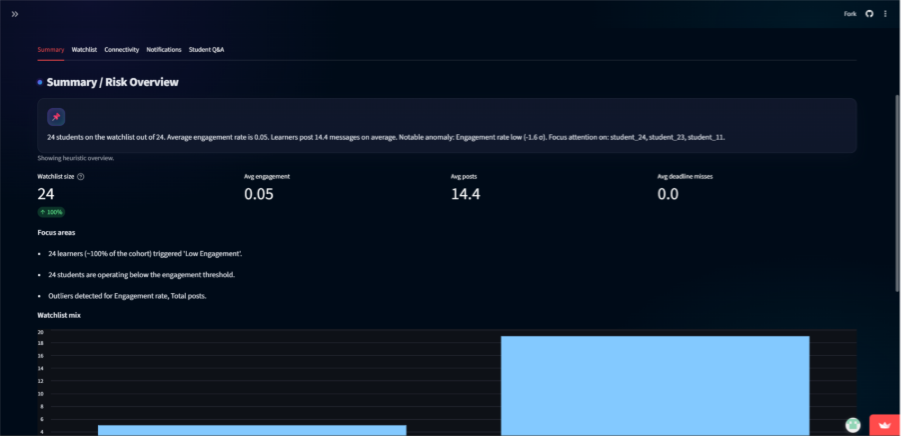


3.5.1. Using the AI Insights Interface

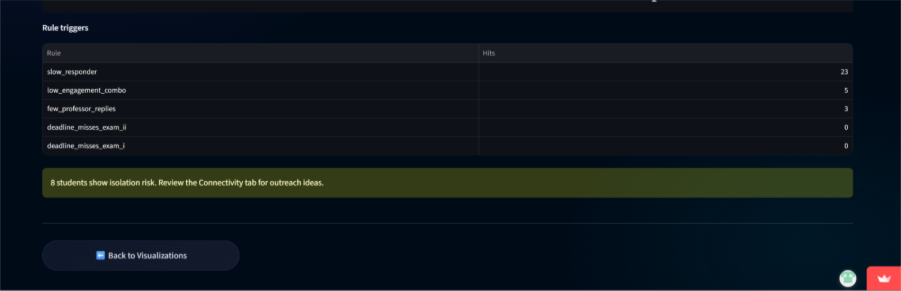
The page presents its findings through a series of tabs designed for specific workflows.

**Summary Tab**

* **What You'll See:** The generated cohort summary, key metric tiles (watchlist size, average engagement), focus bullet points, triggered rules, and a "watchlist mix" chart.

*Figure 37: The Summary Tab header showing the AI-generated narrative text.*

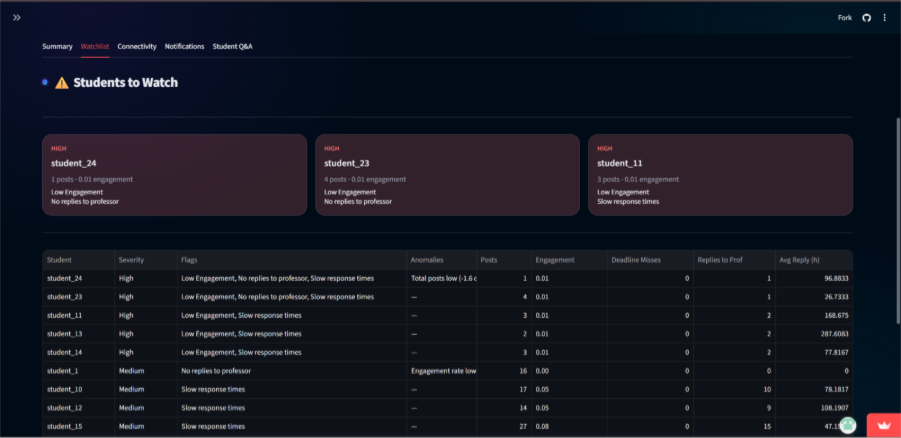
*Figure 38: The key metrics tile dashboard displaying watchlist size, avg engagement, average posts, etc.*

*Figure 39: The "watchlist mix" bar chart visualizing the distribution of students by severity level.*

* **Action Steps:**
  + Read the summary for a high-level overview.
  + If warnings about missing columns appear, check your OAM headers and re-run Attribute Analysis.
  + Proceed to the **Watchlist Tab** for detailed intervention planning.

**Watchlist Tab**

* **What You'll See:** Prioritized, scannable student "cards" color-coded by severity, a detailed watchlist dataframe, and suggested playbook actions.

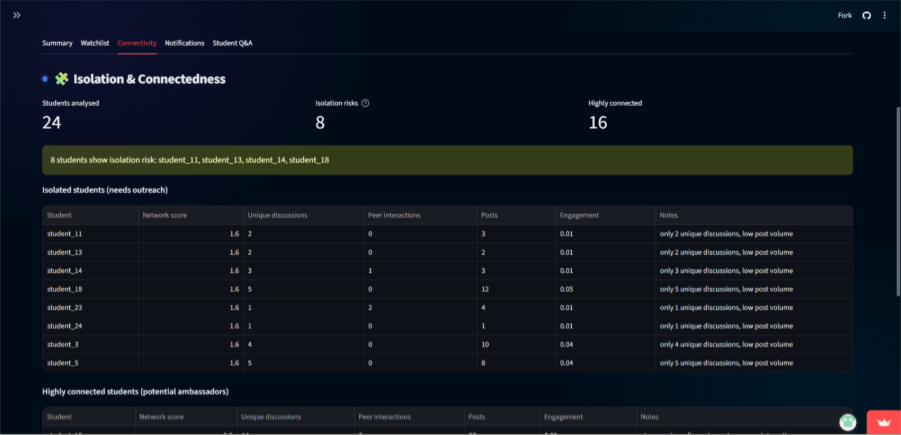
*Figure 40: The Watchlist Tab view showing severity-coloured student cards and the detailed data frame.*

*Figure 41: A close-up of a high-severity student card with flags, anomaly details, and suggested playbook actions.*

* **Action Steps:**
  + **Review high-severity (red) entries first.**
  + Use the provided **playbook actions** as templates for your first outreach.
  + **Export the watchlist table** to include in assessment appendices or intervention logs.

**Connectivity Tab**

* **What You'll See:** Counts of "Isolated" and "Connected" students, sample lists (up to 20 each), and the median thresholds used for classification.

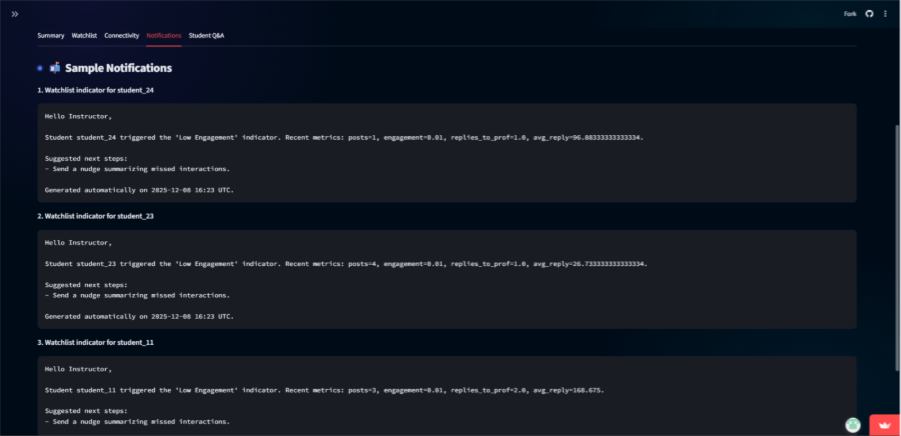
*Figure 42: The Connectivity Tab main view showing the "Isolated" and "Connected" student lists.*

*Figure 43: The panel displaying the median thresholds and weighted scoring formula used for classification.*

* **Action Steps:**
  + **Target isolated students** for supportive outreach (e.g., direct messages, office hour invitations).
  + **Consider recruiting connected students** as peer mentors or discussion facilitators.

**Notifications Tab**

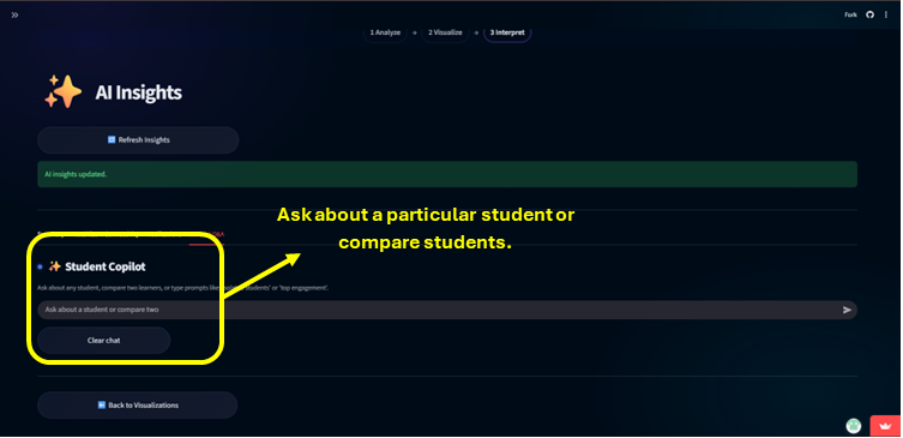
* **What You'll See:** Up to 3 sample, instructor-ready email drafts (subject and body) generated for watchlisted students.

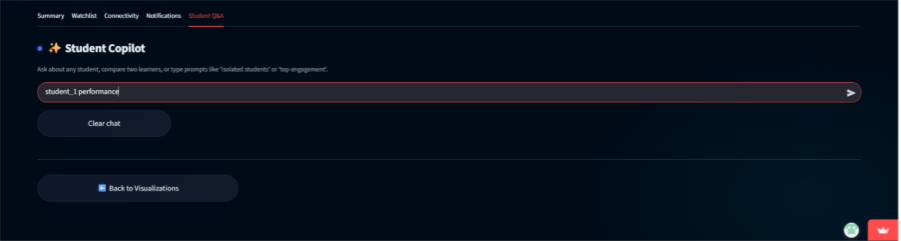
*Figure 44: The Notifications Tab displaying sample email templates ready for copy-paste.*

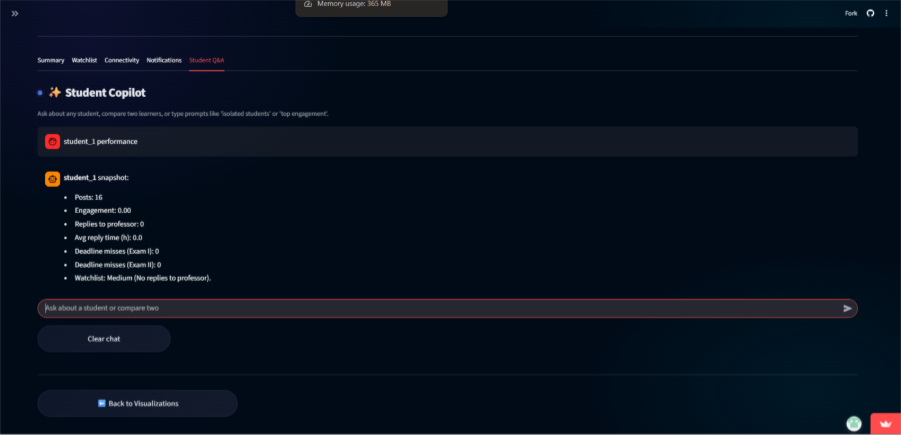
* **Action Step:** **Copy and paste** these templates into your LMS or email client. Customize them as needed.

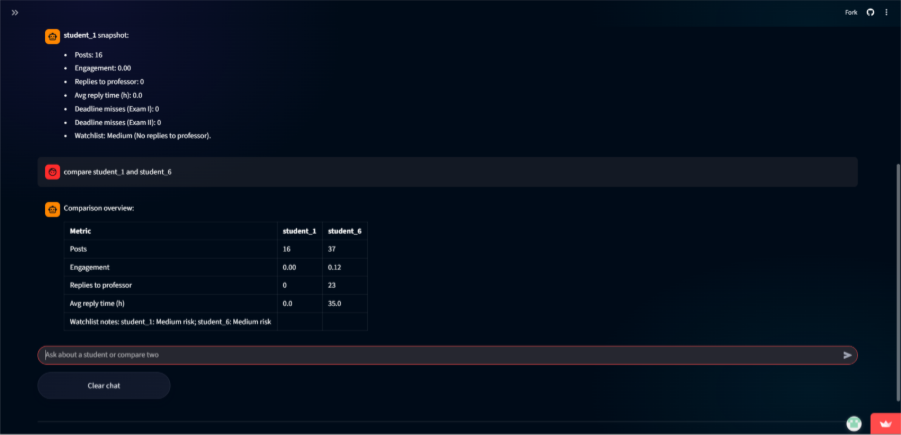
**Student Q&A (Copilot)**

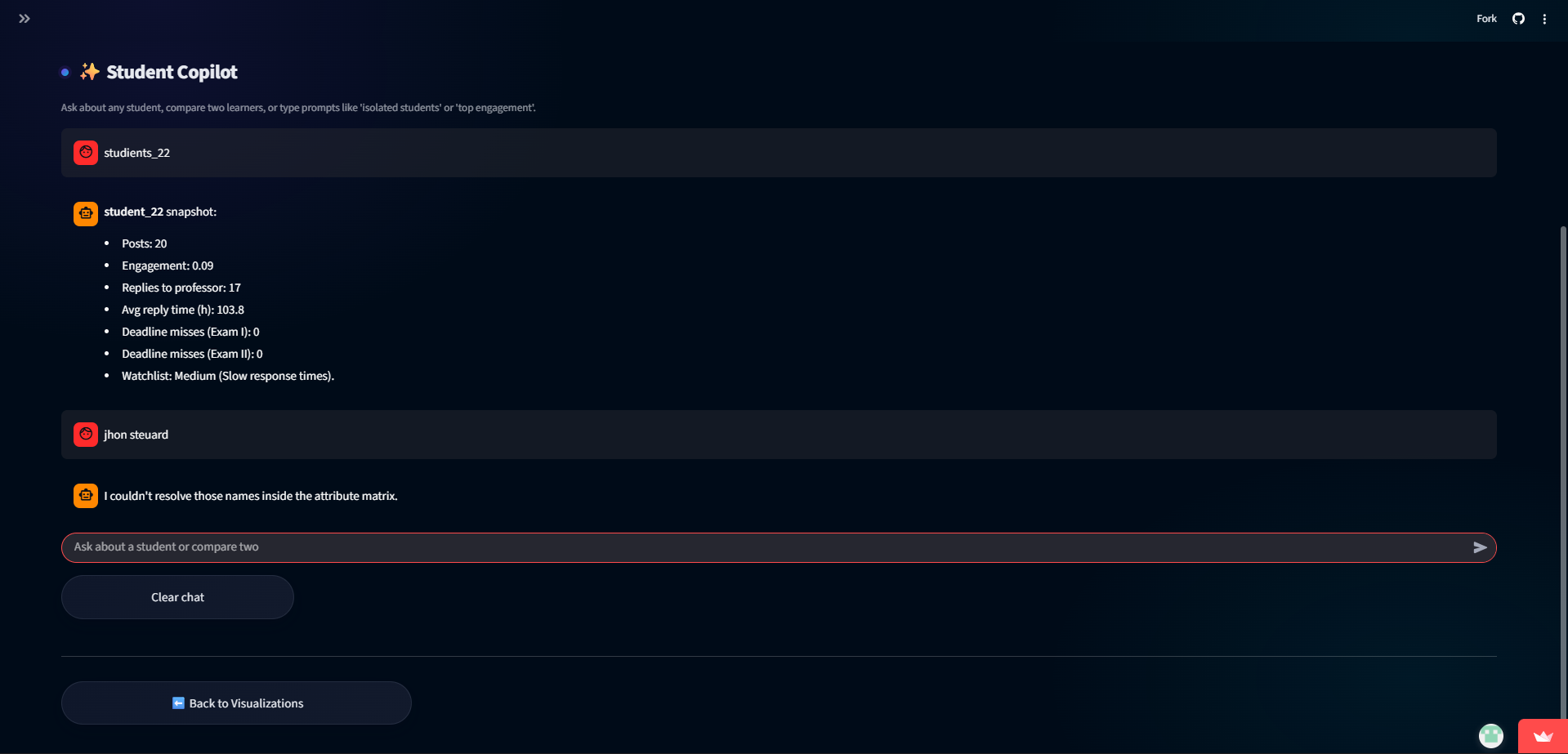
* **What It Is:** A chat-based assistant that answers natural language questions about students and cohort metrics.

*Figure 45: The Student Q&A Copilot interface with the chat input box.*

*Figure 46: Example of a query.*

*Figure 47: Example of a query response showing a single-student snapshot.*

*Figure 48: Example of a query response comparing two students in a markdown table.*

*Figure 49: A system message warning about fuzzy name matching limitations.*

* **Capabilities:**
  + Provide a snapshot for a single student (posts, engagement, watchlist status).
  + Compare two students in a summary table.
  + Answer cohort-level queries: *"Who are the most isolated students?"*, *"Show top 5 by engagement."*
* **Limitation:** Name matching uses fuzzy logic. **Always verify** the identity of students returned for ambiguous queries (Figure 49).
* **Use Case:** Quickly prepare for student meetings or generate comparative notes during grading deliberations.

# **4. Exports, Session History & Collaboration**

The Moodle Log Analyzer is designed for transparent, reproducible analysis. All data processing occurs locally, and you maintain full control over exporting and sharing artifacts for collaboration, auditing, or documentation.

**Exporting Artifacts**

You can export every major output of the pipeline for offline review, sharing, or inclusion in reports. Each page provides relevant export functionality.

|  |  |  |  |
| --- | --- | --- | --- |
| **Export Type** | **Source Page** | **Format** | **Use Case** |
| **Processed Raw Dataset** | **Data Upload** | CSV | Share the cleaned, normalized version of your original upload. |
| **Object Attribute Matrix (OAM)** | **Attribute Analysis** | CSV | Share the complete computed metrics for all students. Essential for external analysis. |
| **Ranking Results** | **Ranking** | CSV | Export the ranked list with overall and per-attribute ranks. |
| **COCO & Validation Tables** | **COCO Analysis** | CSV | Export the fairness validation results. |
| **Configuration Snapshot** | **Configuration** (Export/Import tab) | JSON | Save all settings (professors, deadlines, rules) to replay an analysis identically. |
| **Visualization Charts** | **Visualization** | PNG | Include charts in presentations or reports. |
| **AI Summary & Watchlist** | **AI Insights** | CSV/Text (via copy) | Export the prioritized watchlist and summary for intervention tracking. |

# **5. Troubleshooting & Quality Gates**

|  |  |  |
| --- | --- | --- |
| **Issue / Symptom** | **Likely Cause** | **Recommended Action** |
| **Upload fails with “Missing required columns”.** | Source export lacks userid, userfullname, or message columns (or naming differs). | Edit the CSV/XLSX headers in Moodle export to match expected names, then re-upload. |
| **Upload spinner never completes or dataset preview shows garbled text.** | File uses uncommon encoding or separator that pandas cannot infer. | Ensure UTF-8/UTF-8-SIG encoding; save as comma-separated CSV from Moodle before retrying. |
| **Excel ingestion error referencing openpyxl.** | openpyxl not installed in the Python environment. | Install dependency via “pip install openpyxl” or use CSV export instead. |
| **Sample data download buttons show warning “Demo CSV/XLSX not found”.** | sample\_data folder missing bundled files or path moved. | Restore discussion\_demo files under sample\_data/ or update file paths inside Data Upload page. |
| **Configuration import fails with JSON error.** | Uploaded configuration file malformed or contains unsupported keys. | Validate JSON syntax, ensure same schema as ConfigManager.to\_dict(), then re-import. |
| **Deadline editor does not save new exams.** | Exam name blank/duplicate or invalid date format entered. | Provide unique exam labels and valid dates; duplicates are ignored by ConfigManager. |
| **“Compute Attributes” raises warnings and columns display zeros.** | Required columns for selected attribute missing or ML packages (transformers, sentence-transformers, textstat) unavailable. | Deselect unsupported attributes or install missing packages; verify dataset contains created/parent/subject fields. |
| **Attribute computation unexpectedly slow or crashes.** | Large dataset with heavy NLP features enabled (topic relevance, AI detection). | Disable ML attributes in configuration or process subsets before recomputing full cohort. |
| **“Run Student Ranking” button disabled or results empty.** | student\_attributes not computed or no attributes selected for ranking. | Return to Attribute Analysis, compute OAM, ensure selected\_attributes is non-empty, then rerun ranking. |
| **Ranking output shows NaN ranks or unexpected sort order.** | Attributes contain all-null values (e.g., due to earlier computation failures). | Re-compute attributes after resolving warnings; inspect data preview for completeness. |
| **COCO matrix preview blank or request returns empty tables.** | No numeric rank columns passed to COCO or outbound HTTPS blocked. | Review ranked dataframe to ensure \*\_rank columns exist; retry on network with COCO access. |
| **“Validation skipped” callout persists.** | Parsed COCO tables missing Delta/Tény and Becslés columns required for validation. | Adjust COCO run or manually add required columns before rerunning validation. |
| **Visualisation tabs show warnings “Attributes/Ranking/COCO missing”.** | Prerequisite datasets not cached in SessionDataManager (e.g., server restarted). | Re-run upstream steps (Attribute Analysis, Ranking, COCO) to repopulate session state. |
| **AI Insights summary defaults to heuristic text with “manual\_mode” note.** | Configured transformer model unavailable or download failed. | Switch to Manual summary in settings or install the selected Hugging Face model locally. |
| **AI Insights lists many “missing\_columns” warnings.** | Dataset lacks metrics referenced in ai\_insight\_rules or thresholds (case mismatch). | Review rule JSON and dataset columns; rename fields or adjust rules to match computed attributes. |
| **Notifications/watchlist empty despite known issues.** | Thresholds too strict or rules disabled; ai\_insights\_dirty flag not reset. | Lower thresholds, review rule severity, and ensure SessionDataManager.store\_ai\_insights is invoked after recomputation. |
| **Transformers/textstat import errors appear during attribute computation.** | Heavy NLP libraries not installed (requirements not fulfilled). | Run “pip install -r requirements.txt” and restart Streamlit; optionally disable ML attributes in configuration. |
| **Analysis history missing steps after refresh.** | Session state cleared (Streamlit restart/tab reload). | Avoid refreshing mid-run; export intermediate artifacts or persist config/history externally. |

# **6. Appendices**

## 6.1. Appendix A – Attribute Dictionary

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Category** | **Description** |
| **total\_posts** | Activity | Count of posts made by a student. |
| **active\_days** | Activity | Unique days with at least one post. |
| **average\_posts\_per\_day** | Activity | total\_posts divided by active\_days. |
| **max\_streak** | Activity | Longest consecutive posting streak. |
| **modification\_count** | Activity | Edits detected via modified timestamp. |
| **total\_replies\_to\_professor** | Engagement | Replies targeting professor posts. |
| **unique\_interactions** | Engagement | Distinct peers a student interacted with. |
| **unique\_discussions** | Engagement | Number of subjects/discussions joined. |
| **engagement\_rate** | Engagement | Replies to professor divided by total professor posts. |
| **avg\_reply\_time** | Engagement | Average hours between parent and child posts. |
| **valid\_response** | Engagement | Flag when response satisfies heuristics. |
| **total\_characters** | Content | Sum of message characters per student. |
| **total\_words** | Content | Sum of token counts per student. |
| **citation\_count** | Content | Occurrences of citation-like patterns. |
| **topic\_relevance\_score** | Content | Semantic similarity to prompts via transformers. |
| **avg\_AI\_involvedMsg\_score** | Content | Average AI-likeness rating (0–10). |
| **deadline\_exceeded\_posts\_Quasi\_exam\_I** | Exam | Posts after Exam I deadline. |
| **deadline\_exceeded\_posts\_Quasi\_exam\_II** | Exam | Posts after Exam II deadline. |
| **deadline\_exceeded\_posts\_Quasi\_exam\_III** | Exam | Posts after Exam III deadline. |
| **Pattern\_followed\_quasi\_exam\_i** | Exam | Counts posts referencing configured parent IDs. |

## 6.2. Appendix B – Configuration Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Configuration Section** | **Setting Name** | **Default Value** | **Description / Function** |
| **Professors** | Professor Names | professor\_1,professor\_2 | A list of usernames that identifies teaching staff. These users are **excluded** from student performance analysis metrics to prevent skewing class averages. |
| **Exam Deadlines** | Exam Dates | *Quasi Exams I, II, III (Nov 2024)* | Defines the cutoff dates for specific assessments. These dates are used to calculate "Deadline Misses" and flag late submissions. Users can add new exams or delete existing ones. |
| **Advanced Settings** | Parent IDs for Pattern Matching | 163486 | A specific set of ID numbers used to detect specific discussion patterns or thread types within the dataset. |
|  | Enable ML-based Attributes | True | Toggles advanced Machine Learning features. When enabled, it computes topic relevance and AI content detection. **Note:** Disabling this may improve performance but reduces analytical depth. |
|  | Y Value for Ranking | *(User Input)* | A numeric reference value used specifically for COCO analysis ranking calculations. |
| **AI Insights** | Summary Model | manual | Selects the AI model used to generate student narratives. Options include "Manual" (fastest, no download) and various Hugging Face models (e.g., FLAN-T5, BART) which require download time. |
|  | Summary Temperature | 0.0 | Controls the "creativity" of the AI summary. Lower values (near 0) result in more deterministic, factual summaries; higher values introduce more variability. |
|  | Low Engagement Threshold | 0.25 | Students whose calculated engagement rate falls below this percentage (0.0–1.0) will be flagged for "Low Engagement". |
|  | Low Posts Threshold | 5 | The minimum number of posts expected. Students with fewer posts than this count are flagged. |
|  | Deadline Miss Threshold | 1 | The allowable margin for late submissions. Students exceeding this number of missed deadlines **per exam** are flagged. |
|  | Replies to Professor Threshold | 1 | Students who have replied to the professor fewer times than this count will be flagged for low interaction. |
|  | Slow Reply Threshold | 24hours | The maximum expected average response time. Students whose average reply time exceeds this duration are marked as "Slow Responders". |
|  | Z-score Thresholds (Low/High) | -1.5/2.5 | Statistical boundaries for identifying outliers. Students with activity Z-scores below -1.5 are low outliers; those above 2.5 are high activity outliers. |
|  | AI Insight Rules (JSON) | *(Predefined Rules)* | A JSON editor defining the logic (playbooks) for tagging students. It combines the thresholds above to assign tags like "Low Engagement" or "Deadline Misses" and suggests actions (e.g., "Send a nudge"). |

## 6.3. Appendix C – Glossary & Keyword Coverage

**Moodle Log Analyzer** – A Streamlit-based web application that automates the evaluation of student participation in Moodle discussion forums by converting raw logs into analytical insights and intervention-ready reports.

**Object Attribute Matrix (OAM)** – The central computed dataset where each row represents a student and each column is a calculated performance metric (attribute). It serves as the foundational data for all ranking, visualization, and AI insight generation.

**COCO Y0 Analysis** – “A component-based object comparison for objectivity” a method that perform antidiscriminative optimization based on similarity analysis.

**AI Insights** – A major feature module that synthesizes cohort data, detects outliers, and generates a prioritized watchlist with actionable narratives and communications using heuristic rules and optional generative AI.

**Watchlist** – A critical output of the AI Insights module: a prioritized list of students flagged for instructor intervention, categorized by severity and accompanied by suggested playbook actions.

**Social Connectedness Score** – A weighted composite metric (derived from attributes like unique\_discussions and unique\_interactions) that quantifies a student's integration within the forum network to classify them as "Connected" or "Isolated."

**Severity Levels** – A key classification system (High/Medium/Low) used to prioritize students on the watchlist based on the number and type of rule violations and anomalies detected.

**ConfigManager** – A core application component that centralizes the management, storage, and validation of all user-defined settings (professors, deadlines, rules) to ensure consistency across the analysis pipeline.

**SessionDataManager** – A core application component responsible for caching user data, computed attributes, rankings, and results within the user's browser session to maintain state and enable efficient data flow between pages.

**Y Reference Value** – A fundamental parameter (default: 1000) in the COCO Y0 algorithm that establishes a standardized baseline performance index. This positive starting point ensures all subsequent fairness calculations avoid negative values, which are incompatible with the required linear programming methods, while providing sufficient numerical granularity for precise ranking and validation outputs.

**Stair Value** – A configuration parameter on the COCO Y0 Analysis page that defines the cohort size or the number of students to be evaluated in a single fairness-validation step.

**AI Insight Rules** – A configuration element, typically defined in JSON, that contains the logical conditions and thresholds (e.g., low engagement, missed deadlines) used to automatically flag students and assign severity levels.

**Playbook Actions** – Actionable, instructor-ready recommendations generated by the system for each watchlisted student, providing concrete first steps for intervention based on the specific flags detected.

**Connected/Isolated Students** – Important classifications derived from the Social Connectedness Score. Connected students are well-integrated (potential mentors), while Isolated students show low network integration (primary outreach targets).

**Statistical Anomaly** – A key detection method using Z-score analysis to identify students whose activity (e.g., very low engagement or very high deadline misses) falls outside typical statistical boundaries, flagging them for review.

**Fairness Validation** – The core conceptual process of algorithmically testing student rankings for bias and consistency, primarily executed through the COCO Y0 analysis and its delta inversion checks.

**Reproducibility** – A key principle of the system ensuring that using the same input data and configuration (saved as a JSON file) will yield identical analytical outputs, which is critical for auditing and academic assessment.

**Attribute Categories** – The organizational framework grouping all computed metrics into four logical themes: Activity (posting behavior), Engagement (interaction quality), Content (message characteristics), and Exam (deadline-related metrics).

**Student Copilot** – A feature component within AI Insights providing a chat-based interface for natural language queries about individual students, comparisons, and cohort-level metrics.

**Moodle Discussion Export** – The primary input to the system: a CSV or XLSX file downloaded from a Moodle forum containing columns such as userid, message, and timestamp fields.

**Configuration JSON** – A key artifact that is a portable snapshot of all analysis settings. It enables perfect reproducibility and sharing of analysis scenarios by capturing professors, deadlines, rules, and thresholds in a single file.

**Streamlit Application** – The technology stack and framework used to build the Moodle Log Analyzer, enabling it to run as an interactive web application with a Python backend and a reactive frontend interface.

**Analysis Pipeline** – The complete, sequential process description of the system, encompassing data upload, configuration, attribute computation, ranking, fairness validation, visualization, and AI-driven interpretation.

**Deterministic Pipeline** – A critical quality attribute of the system guaranteeing that the same inputs and configuration will always produce identical outputs, ensuring transparent and auditable evaluation results.

**Sanitized Datasets** – Processed versions of the original Moodle export where personally identifiable information (PII) has been removed or pseudonymized. These datasets enable safe collaboration and sharing of analysis scenarios without compromising student privacy.

**Auditable Evidence** – The complete set of exported artifacts—configuration JSON, processed data, rankings, validation results, and session logs—that provides a transparent, step-by-step record of the analysis. This evidence is essential for peer review, grading justification, and research reproducibility.