PyCAT

Python Chemometric Agile Tool

Technical Documentation

Version	1.0
Date	28 July 2026
Author	Dr. Emanuele Farinini, PhD
Status	Confidential – Proprietary
Copyright	© 2026 Dr. Emanuele Farinini, PhD
Licence	All rights reserved. Access via personal licence code only.

1. Project Description

PyCAT (Python Chemometric Agile Tool) is a proprietary, web-based chemometric platform developed entirely in Python by Dr. Emanuele Farinini, PhD. It delivers the full power of advanced chemometric methods through a modern, interactive browser-based interface — with no installation required by the end user.

PyCAT represents the evolution of the CAT (Chemometric Agile Tool) ecosystem, originally developed in R by Prof. R. Leardi, C. Melzi and G. Polotti, and widely adopted in academic and industrial settings worldwide. PyCAT preserves the rigorous methodological foundation of CAT while introducing a modern Python stack, interactive Plotly visualisations, and a web-deployable Streamlit architecture.

PyCAT exists in two distinct modes of use: a **public Demo version** and a **full professional version** used internally for consulting engagements.

	Demo Version	Full Version (Consulting)
Access	Free, upon registration (personal licence code via email)	Used directly by Dr. Farinini in client engagements
Purpose	Allows users to explore the workflow, test modules and experience the interface	Used to deliver chemometric analyses as part of consulting projects
Data	Users upload their own datasets	Client data processed by the author under confidentiality agreement
Audience	Researchers, students, industry professionals interested in chemometrics	Pharmaceutical, food, chemical and environmental industry clients
Output	Interactive results, downloadable reports	Full analytical reports, models and recommendations

2. Purpose and Scope

2.1 Scientific Purpose

- Rigorous, validated implementation of core chemometric methods: PCA, PLS, MLR-DoE, MCR-ALS, NWAY, classification, GA variable selection, image analysis
- Make advanced multivariate analysis accessible to non-programmers through an intuitive GUI
- Support the full analytical lifecycle: data import → pre-treatment → modelling → validation → export
- Excellent teaching tool for chemometrics courses at university and industrial level

2.2 Demo Version — Purpose and Strategy

The Demo version is publicly accessible after a short registration (name and valid email address). Its primary purposes are:

- Allow researchers, students and industry professionals to experience the PyCAT workflow and interface firsthand
- Serve as a teaching and training tool for chemometrics courses

- Generate qualified leads: registration data allows the author to identify and follow up with potential consulting clients
- Demonstrate the platform's capabilities to prospective clients before a consulting engagement

2.3 Full Version — Consulting Use

The full, unrestricted version of PyCAT is used exclusively by Dr. Emanuele Farinini, PhD in the context of professional chemometric consulting projects. It is not distributed. Clients receive analytical results, models and reports — not access to the software itself.

- Applied to real client datasets under confidentiality agreement
- Enables delivery of PCA, PLS, DoE, classification, image analysis and custom workflows
- Integrated into consulting projects for pharmaceutical, food, chemical and environmental industries

---

3. Software Architecture

3.1 Technology Stack

Layer	Technology	Role
Frontend	Streamlit (Python)	UI rendering, interactive widgets, sidebar navigation
Backend	Python 3.11	All computation, data processing, modelling
Numerics	NumPy / SciPy	Matrix operations, statistical functions, optimisation
Data	Pandas	Dataset management, I/O, transformations
Visualisation	Plotly	Interactive charts (2D/3D, heatmaps, surface plots)
Image proc.	OpenCV / PIL	Image loading, masking, colour-space conversion
Deployment	Streamlit Cloud	HTTPS web hosting, session isolation, access control
Auth	Custom (auth_utils)	Personal licence code validation (PY-XXXXXXXX format)

3.2 Application Structure

The application follows a page-based routing architecture. The entry point `streamlit_app.py` configures the Streamlit session and delegates to `homepage.py`, which renders the sidebar navigation and dispatches rendering to the appropriate page module based on `st.session_state.current_page`.

- Each analytical module is a self-contained Python file (*_page.py) with a show() entry point
- Utility sub-packages (*_utils/) encapsulate calculations, plots and I/O separately from UI code
- Workspace state (datasets, models, results) is persisted across pages via Streamlit session state
- Authentication enforced at startup via personal licence code

3.3 Data Flow

1. User uploads data (CSV / Excel / TXT) in Data Handling → stored in session state. 2. Pre-treatments applied in Transformations → stored as a new dataset. 3. Analytical modules read from the active dataset.

4. Results displayed interactively and exportable to Excel / PNG.

---

4. Licence and Intellectual Property

PyCAT is proprietary software. All source code, algorithms, UI design, documentation and associated materials are the exclusive intellectual property of Dr. Emanuele Farinini, PhD.

- © 2026 Dr. Emanuele Farinini, PhD. All rights reserved.
- Access is granted exclusively via a personal licence code issued upon registration.
- Unauthorised copying, redistribution, reverse engineering or modification is strictly prohibited.
- The software may be used solely for the purpose and scope agreed in the licence agreement.
- To cite: Farinini, E. (2026). PyCAT: Python Chemometric Agile Tool. chemometricsolutions.com

This document constitutes a technical record of the software's existence and design as of the date shown on the cover page, and may be used as evidence of authorship and prior art in any intellectual property dispute.

---

5. How to Cite

> **Farinini, E. (2026).** *PyCAT: Python Chemometric Agile Tool* [Software]. Available at: https://chemometricsolutions.com