

EcoRETINA Workbench PRO Official Guide



Universidad Complutense

June 18th, 2026

Madrid

Jules BERNADAC & Maxime ELEK

1. PLATFORM PHILOSOPHY & AVAILABLE ALGORITHMS	3
1. Selective Non-Linear Approximation (Eco-RETINA):	3
2. Parametric Shocks & Regularization (OLS, Lasso, Ridge, ElasticNet):	3
3. Complex Non-Parametric Frameworks (XGBoost, Random Forest, Neural Network):	3
2. GRAPHICAL USER INTERFACE GUIDE (TAB-BY-TAB)	3
Tab 1: Data & Pre-Processing (Import & Cleansing)	4
1. Dataset Import:	4
2. Global Configuration:	4
3. Outlier Management:	4
4. String / Categorical Variables:	4
5. Drop Any Variable:	4
6. Dynamic Grid Visualizer:	4
Tab 2: Algorithms & Params (Strategy Execution)	4
• When using Eco-RETINA:	4
• When using alternative algorithms (OLS, XGBoost, RF, Lasso, etc.):	5
Tab 3: Compare Results (Benchmark Board & Diagnostics)	5
• The Comparison Table:	5
• The Advanced Econometric Suite (View Statistical Report):	5
1. Statistical Report:	5
2. Visual Formula:	5
3. Graphical Analysis:	5
Tab 4: Predict (Operational Deployment)	6
1. Active Model Selection:	6
2. Operational Workloads:	6
○ Single Evaluation (Interactive Form):	6
○ Batch Processing Engine (CSV Import):	6
3. MANDATORY ANTI-CRASH MANUAL & BEST PRACTICES	6
1. Target Leakage:	6
2. The Zero-Value Trap (Eco-RETINA only):	6
3. Framework Cross-Over Errors:	7
4. FINAL READINESS CHECKLIST	7
• Target Clarity:	7
• Numeric Matrix:	7
• Leakage Purged:	7
• Dummy Traps Avoided:	7
• Data Validation:	7

1. PLATFORM PHILOSOPHY & AVAILABLE ALGORITHMS

The **EcoRETINA ML Workbench PRO** is an advanced, comprehensive benchmarking platform that bridges the gap between traditional structural econometrics and state-of-the-art predictive Machine Learning. Instead of restricting analysis to a single technique, it allows users to compare different mathematical philosophies side-by-side to achieve optimal out-of-sample predictive power.

The application natively integrates **8 major algorithms** divided into three strategic families:

1. Selective Non-Linear Approximation (Eco-RETINA):

Automatically generates thousands of mathematical input transformations (powers, polynomial expansions, and cross-variable interactions) and isolates the most robust subset. This is the ideal tool for discovering hidden functional or physical laws.

2. Parametric Shocks & Regularization (OLS, Lasso, Ridge, ElasticNet):

Excellent for isolating exact individual elasticities and marginal effects while applying L1/L2 mathematical penalties to prevent model overfitting.

3. Complex Non-Parametric Frameworks (XGBoost, Random Forest, Neural Network):

Engineered to automatically map high-order non-linearities and deep feature interactions without requiring manual data conversion or initial variable transformations.

2. GRAPHICAL USER INTERFACE GUIDE (TAB-BY-TAB)

The application structures your data science workflow into 4 sequential steps, accessible through the top navigation tabs.

Data & Pre-processing	Algorithms & params	Compare results	Predict
-----------------------	---------------------	-----------------	---------

Tab 1: Data & Pre-Processing (Import & Cleansing)

Before choosing an algorithm, data integrity must be flawless.

1. Dataset Import:

Load your raw dataset. The platform automatically handles delimiter (, or ;) and encoding (UTF-8 or Latin-1) detection to prevent parser crashes.

2. Global Configuration:

Select your dependent column (*Target Y*) and adjust the *Train Split (%)* slider to isolate your validation sets (e.g., 80% Train / 20% Validation).

3. Outlier Management:

Select a continuous variable to evaluate its distribution. Set custom clipping boundaries to either cap values (*Clip*) or purge rows (*Drop*).

4. String / Categorical Variables:

Convert textual or string columns into clear binary groups (0/1) using the *Encode (Dummies)* button, automatically omitting the baseline reference category to prevent the dummy-variable trap.

5. Drop Any Variable:

Instantly drop technical columns (such as index rows or unique identifiers) or variables responsible for **target leakage** (future information exposure).

6. Dynamic Grid Visualizer:

Click "**Visualize Processed Dataset**" to open an interactive data table. This final check ensures that all your rows are entirely numeric and clean before triggering model execution.

Tab 2: Algorithms & Params (Strategy Execution)

This tab allows you to configure your selected algorithm using sub-navigation options:

- When using Eco-RETINA:

You must manually sort your features. Quantitative metrics (e.g., age, dimensions, prices) must be added to the **Continuous Variables** list, while binary conditions (0/1) belong in the **Dummy Variables** list. Input your exponent search list (e.g., [-1.0, 0.0, 1.0]). Use OLS to predict a continuous output, probit/logit for a binary one.

- When using alternative algorithms (OLS, XGBoost, RF, Lasso, etc.):

The workbench automatically routes all selected features to the model. You get absolute control over hyperparameter tuning (number of estimators, tree depth, learning rate init, penalty Alpha weights, or multi-layer hidden network shapes).

Click ► **Run Active Pipeline** to trigger training. The routine executes on a background worker thread, ensuring the workbench remains smooth and responsive.

Tab 3: Compare Results (Benchmark Board & Diagnostics)

The nerve center of the application. Every completed pipeline registers an historical summary row.

- The Comparison Table:

Instantly tracks statistical accuracy (R^2 Train/Test, RMSE, MAPE) against ecological costs (carbon footprint in kg of CO₂eq computed via CodeCarbon). The most efficient metrics are **tagged with an excellence star (★)**.

- The Advanced Econometric Suite (View Statistical Report):

Right-click any log row or click “view statistical report” to open an in-depth analytical window split into 3 segments:

1. *Statistical Report:*

For linear setups and Eco-RETINA, it displays exact weights, the Shapiro-Wilk residuals test, as well as **Marginal Effects** and **Mean Elasticities**. For non-parametric tree models and neural networks, it renders a structured relative **Feature Importance** breakdown.

2. *Visual Formula:*

Dynamically compiles the trained numerical weights into an easily readable algebraic format (e.g., $Y=0.5+3.2\times X1-0.15\times X22$).

3. *Graphical Analysis:*

Displays three Matplotlib figures: *Actual vs Predicted* (including a **translucent shaded 95% Confidence Interval band**), an error histogram with a live normality *p-value*, and a forest plot mapping confidence boundaries for parametric coefficients.

Tab 4: Predict (Operational Deployment)

This tab unlocks your models' value by deploying them to perform work on unseen records.

1. Active Model Selection:

Select your optimal run validated from the benchmark log sheet (e.g., [Run_XGBoost_142205](#)).

2. Operational Workloads:

- ***Single Evaluation (Interactive Form):***

The user interface dynamically builds custom field inputs mapping the exact parameters required by the selected model. Input individual inputs and click *Predict* for an instantaneous result.

- ***Batch Processing Engine (CSV Import):***

Upload a new unlabeled dataset file. Click *Run Batch Prediction*: the workbench handles all transformations, applies the model architecture, and spits out an updated CSV file appended with a dedicated [PREDICTED_TARGET](#) column.

3. MANDATORY ANTI-CRASH MANUAL & BEST PRACTICES

To squeeze the absolute best performance from the Workbench, avoid these three logical errors:

1. Target Leakage:

Never keep a predictor that acts as a consequence, component, or future realization of your target variable. (Example: If modeling whether someone uses AI ([AI_USE](#)), you must drop [FREQUENCY_OF_AI_USE](#). Otherwise, your accuracy will be artificially perfect but fail completely on real-world tests).

2. The Zero-Value Trap (Eco-RETINA only):

If negative parameters (like [-1.0](#)) are active in your settings, Eco-RETINA drops all observations containing a 0 value inside your *Continuous Variables* list to protect the math engine from division-by-zero errors.

- *Best Practice:* Never place a binary dummy variable (0/1) inside the continuous list. If your numerical columns naturally contain zeros, clear out negative exponents from your parameter inputs or translate your data so you don't have any 0 anymore. A major upgrade of Eco-Rétina could be to detect if a continuous variable contains some zeros and change the exponent parameter only for this variable.

3. Framework Cross-Over Errors:

Never pick **logit** or **probit** regression types if your Target (Y) is a continuous quantitative value (like income or wage). The target must be strictly binary (0 or 1 only).

- Do not run linear OLS models on purely categorical classifications unless they have been binarized beforehand.

4. FINAL READINESS CHECKLIST

Your file and workbench configuration are ready for optimal execution once you can confirm "Yes" to all five steps:

- **Target Clarity:**

The dependent target column is cleanly isolated and maps your objective.

- **Numeric Matrix:**

All selected predictors are strictly numeric (no text strings left).

- **Leakage Purged:**

All tracking keys (**ID**, **Row_Number**) and downstream consequence features have been filtered out using the *Drop Any Variable* control.

- **Dummy Traps Avoided:**

Categorical rows have been expanded into dummy indicators, explicitly omitting one category out as the reference anchor.

- **Data Validation:**

The final mathematical structure has been validated and checked using the *Visualize Processed Dataset* data viewer.

N.B. : This manual is a first draft. Feel free to suggest upgrades in terms of contents or presentation. The GUI is also a recent upgrade so unexpected behaviours can happen, contact So it can be treated.