

# Local Web-Based CSV Machine Learning Tool

Foundations of Machine Learning, Centrale Méditerranée

Xufeng Zhang, xufeng.zhang@inria.fr

## Objective

Build a locally runnable HTML-based application that allows users to upload CSV files, preview data, and run basic machine learning models using a Python backend. A **demo presentation is required at the end of the course**.

## Requirements

### 1. CSV Upload & Preview

- Provide a file upload interface that accepts **CSV files only**.
- Show an error if the uploaded file is not a CSV.
- Display the **first five rows** (with possible header) for preview.

### 2. Header Detection

- Automatically detect whether the CSV contains a header row.
- Allow the user to manually correct the header detection through simple UI controls.

### 3. Target Column Selection

- Automatically select the **last column** as the output/label.
- Provide a dropdown so users can change the target column.

### 4. Task Type Detection

- Automatically determine whether the task is **regression or classification** based on the target column.
- Allow users to override this selection.

### 5. Model Selection

- Provide a **Descriptive Statistics** button.
- For regression tasks: include at least **three models** (e.g., Linear Regression, Random Forest Regressor, SVR).

- For classification tasks: include at least **three models** (e.g., Logistic Regression, Random Forest Classifier, SVM).
  - Each model available as a clickable button.
6. **Model Training & Output**
- Use Python to train the model with an **80/20 train-test split**.
  - Display evaluation results and relevant plots (e.g., predicted vs actual, confusion matrix).
7. **Technology Requirements**
- Frontend: HTML, CSS, JavaScript with a **clean, modern design**.
  - Backend: Python (e.g., Flask), using pandas, scikit-learn, matplotlib.
  - Must run locally with minimal setup (e.g., python app.py).
8. **AI Assistance**
- You are encouraged to use a large language model to help generate code, refine design, and troubleshoot.
  - You are responsible for integrating and validating all generated content.

## **Deliverables**

1. **Complete Source Code** (HTML/CSS/JS + Python backend)
2. **README** with setup and run instructions (if possible)
3. **End-of-course Demo Presentation**, showing:
  - Uploading a CSV file
  - Header and target selection
  - Task type selection
  - Running at least one regression and one classification model
  - Viewing results and visualizations