Spectral Soil Modeler

Phase 2 Report

Project: The Spectral Soil Modeler

An automated ML web application for predicting soil properties from spectral data

Course: Software Systems Development

International Institute of Information Technology, Hyderabad

Team 27

Contributors:

Anupam Dwivedi	2025201032
Devansh Singh	2025204007
Shobhan Parida	2025201043
Krishna Pokuri	2025202024
Yashwanth B K	2025201028

Date: Wednesday, November 13, 2025

Table of Contents

1. Project Overview

The **Spectral Soil Modeler** is an automated ML web application for soil scientists to predict soil properties from spectral data. The system discovers optimal processing and modeling pipelines by exploring 15 combinations of preprocessing techniques and algorithms.

Core Features:

- **Dual-Mode:** Training Mode (model building) + Prediction Mode (inference)
- **15 Combinations:** 5 ML algorithms × 3 preprocessing techniques
- **5 Algorithms:** PLSR, GBRT, SVR, KRR, Cubist
- **3 Techniques:** Reflectance, Absorbance, Continuum Removal
- **3 Paradigms:** Standard (fast), Tuned (optimized), Both (comparison)
- **Interactive Dashboards:** Results visualization, model ranking, export
- **Complete Persistence:** Model + metadata serialization for reproducibility

2. System Architecture & Workflow

Training Mode Workflow:

1. Upload CSV/XLS spectral data and select target variable
2. Validate data and display statistics
3. Select training paradigm and split ratio
4. Apply 3 preprocessing techniques in parallel
5. Train 15 (or 30) algorithm-technique combinations
6. Compute metrics: R^2, RMSE, MAE, RPD
7. Display interactive dashboard with rankings
8. Save best model with complete metadata

Prediction Mode Workflow:

1. Load previously trained model and metadata
2. Display model details and performance
3. Upload new unlabeled spectral data
4. Apply same preprocessing as training
5. Generate predictions and export results

Key Modules: Data Loading, Preprocessing (3 techniques), Model Training (5 algorithms), Hyperparameter Tuning (Grid/Randomized Search), Evaluation, Persistence, Visualization, UI/Theming, Export, Logging.

3. UI Screens & User Journey

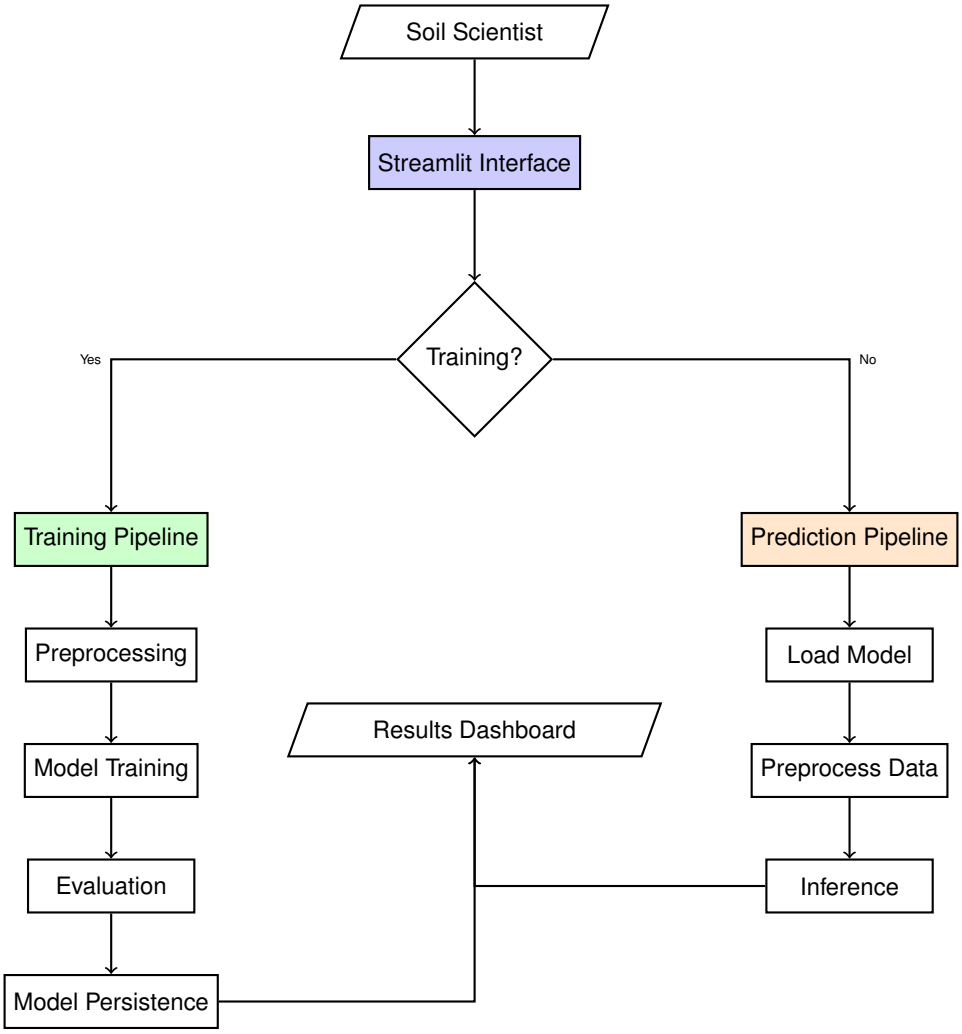
Training Mode: (1) Upload CSV/XLS data $\rightarrow$ (2) Validate and profile data $\rightarrow$ (3) Select target column and training paradigm $\rightarrow$ (4) Configure train/test split $\rightarrow$ (5) Execute training (displays progress bar) $\rightarrow$ (6) View interactive dashboard with KPIs, leaderboards, performance charts, and model comparisons $\rightarrow$ (7) Export results (CSV/Excel/JSON) $\rightarrow$ (8) Automatically save best model and metadata.

Prediction Mode: (1) Load previously trained model from dropdown $\rightarrow$ (2) Display model metadata (algorithm, technique, R^2, hyperparameters) $\rightarrow$ (3) Upload new unlabeled spectral data $\rightarrow$ (4) Apply same preprocessing as training $\rightarrow$ (5) Generate predictions $\rightarrow$ (6) Display results in table and charts $\rightarrow$ (7) Export predictions as CSV.

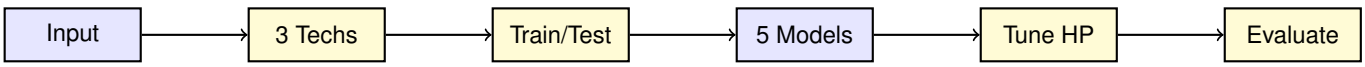
Dashboard Components: Overview tab (KPIs: Best R^2, Mean R^2), Analytics tab (scatter plots: R^2 vs. RMSE, heatmaps), Leaderboard tab (ranked models), Model Details tab (in-depth statistics and hyperparameters).

4. System Architecture Diagrams

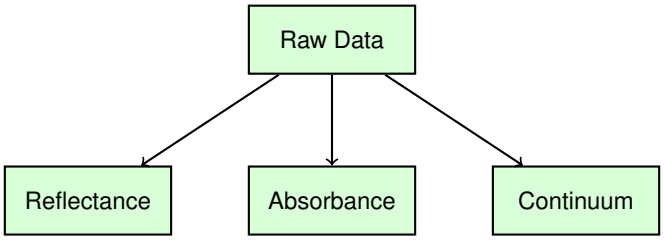
High-Level System Architecture:



Low-Level Data Flow (Pipeline Stages):



Preprocessing Techniques:



Technology Stack: Python 3.12, Streamlit (UI), Scikit-learn (ML), Pandas/NumPy (data), SciPy (signal processing), Plotly (visualization), Joblib (serialization).

5. Machine Learning Approach

Algorithms:

- **PLSR:** Linear, dimensionality reduction, handles multicollinearity
- **GBRT:** Ensemble trees, non-linear patterns, feature interactions
- **SVR:** Kernel-based, robust to outliers, flexible for non-linear
- **KRR:** Kernel ridge regression, smooth predictions, regularization
- **Cubist:** Model trees, interpretable rules, fast inference

Evaluation Metrics:

- **R^2 Score:** Coefficient of determination [0,1], represents variance explained by the model
- **RMSE:** Root mean squared error in target units, penalizes large errors
- **MAE:** Mean absolute error, robust to outliers, intuitive interpretation
- **RPD:** Residual Prediction Deviation (standard deviation / RMSE), indicates predictive capability; >2.0 good, >2.5 excellent

Hyperparameter Tuning: 5-fold cross-validation with Grid Search (exhaustive) or Randomized Search (sampling-based). Search space defined per algorithm with predefined parameter grids.

6. Technology Stack & Implementation

Core Technologies:

- **Python 3.12:** Core language
- **Streamlit:** Web application framework
- **Scikit-learn:** ML algorithms and preprocessing
- **Pandas/NumPy:** Data manipulation and numerical computing
- **SciPy:** Advanced signal processing
- **Plotly:** Interactive visualizations
- **Joblib:** Model serialization

Optional Integration: Google Gemini / OpenAI for AI insights and natural language explanations.

Key Features:

- **Dark/Light Theme:** CSS-based dynamic theming without page reload
- **Interactive Dashboards:** KPIs, leaderboards, scatter plots, heatmaps
- **Model Persistence:** Serialized models + comprehensive JSON metadata

- **Export:** CSV, Excel, JSON formats for integration with other tools
- **Performance Tracking:** Centralized logging and execution monitoring

7. Design Decisions & Future Work

Key Architectural Decisions:

1. **Dual-Mode:** Separates computationally intensive training from lightweight prediction, reducing inference latency
2. **Systematic Exploration:** All 15 algorithm-technique combinations provide robust model discovery without upfront domain assumptions
3. **Configurable Paradigms:** Standard (speed-optimized), Tuned (accuracy-optimized), Both (comprehensive comparison) balances practical trade-offs
4. **Complete Metadata:** Full serialization of preprocessing parameters, hyperparameters, and metrics ensures reproducibility and scientific rigor
5. **Grid/Random Search:** Exhaustive or sampling-based hyperparameter tuning adapted to search space size

Phase 2 Improvements from Phase 1:

- Enhanced UI with dark/light themes and improved visual hierarchy
- Expanded model support (added Cubist and KRR)
- Optimized preprocessing techniques with automated wavelength selection
- Comprehensive evaluation metrics and interactive dashboards

Phase 1 Revisions: Originally single linear workflow $\rightarrow$ dual-mode; basic training $\rightarrow$ 3 paradigms; static plots $\rightarrow$ interactive dashboards; simple export $\rightarrow$ comprehensive metadata persistence; basic UI $\rightarrow$ professional theme system.

Conclusion: The system provides soil scientists with a modular, user-friendly platform for discovering optimal ML pipelines for spectral soil data analysis with full reproducibility and professional visualization capabilities.

Demo Video

You can watch a short demo of the Spectral Soil Modeler (running the notebook and Streamlit dashboard) here:

Spectral Soil Modeler — Demo (YouTube)