Start with Rust ML using Smartcore

Lorenzo Moriondo
tunedconsulting@gmail.com
Github: @Mec-iS Youtube: @tunedlorenzo

August 25, 2025

Minimalistic and pragmatic Machine Learning

Introduction

- I am Lorenzo, one of the contributors at Smartcore.
- I bring a pragmatic perspective on ML dev in Rust.
- APIs that glue Web, Databases, Mobile, IoT... and I love writing Rust code!
- Music and short videos with AI-powered tools.
- Anybody explored the Rust ML ecosystem?
- **Are we learning yet???**

Statistical Learning and Machine Learning

- **Statistical Learning** (see foundational books)
- Before: Smarter data selection due to limited computing power.
  - Smaller datasets, minimal/relevant features (e.g. PCA)
  - Modular-minimal data pipelines (close to C performance, highly debuggable, explainable)
  - Efficient ML on millions of points
- After: Abundant, large-scale datasets and feature-rich pipelines.
- Move up in scale to ML as systems evolve and feedback is collected.
- Smartcore enables both small scale and progressive ML development.

Smartcore: Design Principles

- **What Smartcore is NOT:**
  - Deep Learning or ANN library (for now).
  - Not data science workflow focused; no BLAS/LAPACK/C++ bindings.
- **What it IS:**
  - Pure Rust traits for N-dim matrices (also supports `ndarray`).
  - Low-level as Rust allows; no unsafe.
  - Resilience over feature count.
  - Full WebAssembly support.
  - **Pitch**: "scikit-learn lite" in pure Rust.
  - Accessible for practitioners, not just scientists.

User's Perspective

- Code in Jupyter Notebooks.
- Differences 0.2 vs 0.3 vs 0.4:
  - Dynamic Sized Types, Trait Object & impl Object.
  - Zero-copy matrices, broad target support, version 0.4 is out!
- Example: Custom K-means with initial centroids.
- Type-safe, efficient code; scripted automation.

Contributor's Perspective

- Clustering example: FastPair.
- Theory from papers, porting from Python/C++ (scikit-learn, scipy) to Rust.
- Fundamental algorithm for Hierarchical Clustering.
- Closest pair methods, effective teamwork, and rewarding engineering!

Implementing or porting foundational algorithms takes time, but is highly satisfying.

Implemented and Ready to Use

Supervised:

- Linear methods, Model selection, Bayesian, Tree-based, SVM

Unsupervised:

- Clustering, PCA, Dimensionality reduction, Ensembles

Accessible to users without deep theoretical background, well documented.

Getting Involved

- Good issues for contributors: display improvements, more tests, new notebooks, refactoring, better docstrings, test datasets, probability prediction, hierarchical clustering, new algorithms.
- Code reviewers available, fostering growth!
- **Great opportunity to learn Rust and ML!**

No ANN, No LLM?

- NN may be decision trees
- Computer Vision as clustering problem
- Deep Learning for embedded (DSP in Music)
- LLM with Rust: check llama-rs!
- Smartcore already used in commercial products (e.g. PostgresML)
- Next-gen integration targets: ANN, causal inference, graph algorithms, survival analysis?

Get in touch: `tunedconsulting@gmail.com`

Conclusion

Open Source project planning is about clear engineering principles, not rigid roadmaps.

Contributors bring new skills, and that shapes the library. For professional growth, contribute and help improve Smartcore!

`https://github.com/smartcorelib`

Conclusion

Thank you for your attention!

tunedconsulting@gmail.com

github.com/sponsors/Mec-iS