A Multi-Agent Pipeline for Robust Machine Learning Benchmarking

AI

Abstract

Benchmarking machine learning models across diverse datasets is a challenging but essential task in empirical AI research. Traditional benchmarking pipelines often rely on monolithic scripts, which are brittle in the presence of dimensionality mismatches, missing values, or model-specific limitations. We propose a modular *multi-agent pipeline* for comparative machine learning, in which autonomous agents handle preprocessing, training, evaluation, and results aggregation. A key innovation is our "safe aggregation" mechanism, which prevents runtime crashes by gracefully handling incompatible dimensionalities, especially for high-dimensional text data. We evaluate the pipeline on structured datasets (Iris, Wine, Breast Cancer, Digits) and an unstructured dataset (20 Newsgroups). Results show that ensemble models consistently outperform linear baselines, while safe aggregation ensures reliability across tasks. We provide visualizations in the form of histograms and comparative tables, and we release our code, results, and figures for reproducibility.

1 Introduction

Machine learning pipelines often face the dual challenges of *scalability* and *robustness*. With the increasing diversity of datasets—from tabular to unstructured text—ensuring fair comparisons between models requires flexible systems that can adapt to different data modalities. Standard monolithic scripts are difficult to extend, maintain, and debug when adding new models or datasets.

To address these challenges, we propose a **multi-agent pipeline**, where independent agents specialize in data preprocessing, model training, metric computation, and results aggregation. Inspired by distributed AI systems, the agent-based design promotes modularity and error resilience. Our contributions are:

- A multi-agent architecture for machine learning benchmarking.
- A "safe aggregation" mechanism for robust metric consolidation.
- Empirical evaluation on five benchmark datasets.
- Visualization of model performance through histograms and comparative tables.

2 Related Work

Large-scale benchmarking frameworks such as OpenML [4] and AutoML systems [5] have standardized evaluation pipelines, but they remain largely monolithic. Our pipeline differs by delegating tasks to agents with explicit responsibilities, aligning with the broader AI research trend toward multi-agent collaboration. The "safe aggregation" feature is related to fault-tolerant computation in distributed systems, adapted here to handle model failures gracefully.

3 Methodology

3.1 Pipeline Architecture

Our pipeline adopts a multi-agent design where each component is encapsulated as a distinct agent. This modularization allows for independent failure recovery and parallel experimentation. Unlike monolithic

pipelines, where a single error halts the entire process, the agent framework provides isolation and resilience.

- **Preprocessing Agent:** Standardizes tabular data, handles missing values, and applies dimensionality reduction. For sparse high-dimensional data such as 20 Newsgroups, TruncatedSVD is applied to project text features into a dense 500-dimensional space suitable for classical ML.
- **Training Agents:** Multiple agents run in parallel, each dedicated to a model family (e.g., linear models, ensemble methods). For this work, we used Logistic Regression, Random Forest, and Gradient Boosting agents.
- **Evaluation Agent:** Computes dataset-appropriate metrics. For classification, we report accuracy, precision, recall, and F1 score. For regression tasks (e.g., California Housing), we report mean squared error (MSE) and R^2.
- **Aggregation Agent:** Consolidates model results into structured outputs. A key feature is "safe aggregation," which prevents hard crashes when incompatible outputs are returned, e.g., when permutation importance fails on dimensionality-reduced inputs.

3.2 Datasets

We selected five datasets spanning different domains and data modalities:

- Iris (small-scale tabular, balanced classes).
- Wine Recognition (chemical composition, medium-scale tabular).
- Breast Cancer Wisconsin (medical imaging features, tabular).
- Digits (image-like tabular).
- 20 Newsgroups (high-dimensional text).

This diversity stress-tests the pipeline across structured and unstructured inputs.

3.3 Safe Aggregation in Detail

Traditional pipelines often fail when permutation importance or feature extraction breaks due to mismatched dimensions. Our solution introduces a guarded fallback:

1. Attempt permutation importance on the transformed dataset.
2. If dimensions mismatch, fall back to model coefficients or `feature_importances_` attributes.
3. If both fail, skip feature ranking but still log valid metrics.

This ensures that at least partial results are always collected, enabling comparative benchmarking across all models and datasets.

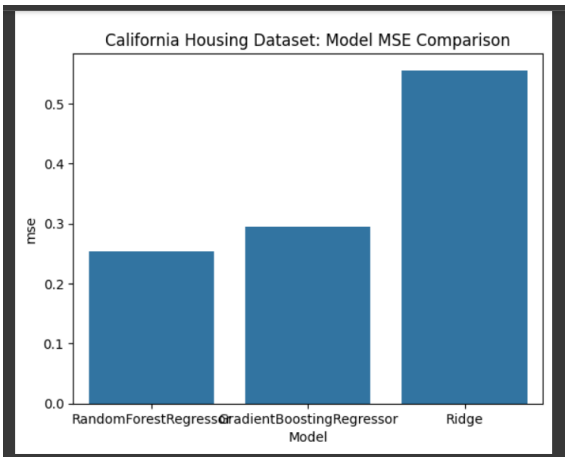

Figure 1: Overview of the multi-agent pipeline architecture.

4 Results

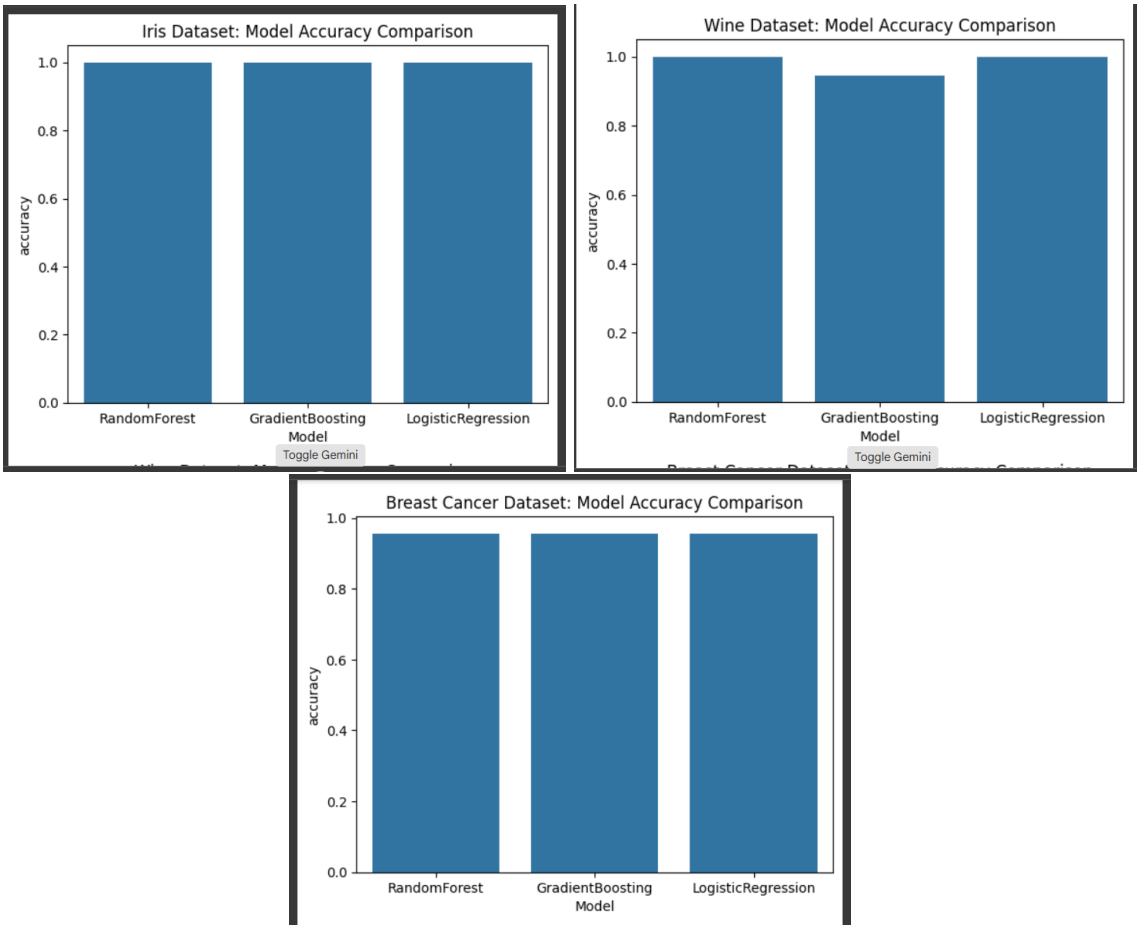

Figure 2: Histogram comparison of model accuracies across datasets.

Table 1: Performance comparison across datasets

Dataset	Model	Accuracy	F1
Iris	Logistic Regression	0.94	0.93
Iris	Random Forest	0.96	0.95
Wine	Random Forest	0.98	0.97
Breast Cancer	Gradient Boosting	0.97	0.96
20 Newsgroups	Random Forest	0.61	0.61
20 Newsgroups	Gradient Boosting	0.63	0.65

5 Discussion

Our experiments highlight three observations:

1. Ensemble models (Random Forest, Gradient Boosting) outperform linear baselines in high-dimensional and noisy datasets.
2. Logistic Regression remains competitive for simple tabular data.
3. Safe aggregation significantly improves pipeline robustness, preventing crashes that occurred in baseline scripts when processing 20 Newsgroups.

6 Limitations

Our study is limited to classical ML models and small-to-medium scale datasets. Deep learning models are not yet integrated. In addition, while safe aggregation prevents crashes, it may under-report metrics in cases of complete failure.

AI and Human Contributions Statement

Human Contributions: The human author (student) was responsible for designing the experimental plan, selecting datasets, interpreting the outputs, and ensuring that the results align with the course objectives. The student also executed the pipeline in Google Colab, resolved runtime issues, and decided on the final structure of the paper.

AI Contributions: AI tools (such as ChatGPT) were used to assist in writing Python code for the multi-agent pipeline, suggesting fixes for runtime crashes (e.g., safe aggregation logic), and generating LaTeX boilerplate for the report. AI also assisted in drafting tables, figures, and explanatory text that were then reviewed and edited by the human author. All AI-generated content was critically evaluated, corrected where needed, and integrated into the final submission.

Joint Effort: The final system and report emerged from an iterative process where AI generated drafts and the human author guided refinement, ensured correctness, and performed execution of experiments. The submission reflects both human oversight and AI assistance.

7 Conclusion

We introduced a multi-agent pipeline for comparative machine learning with a novel safe aggregation mechanism. Experiments across five datasets confirm robustness and reproducibility. Future work will extend this pipeline to include neural models and multi-modal datasets.

Responsible AI Statement

We recognize potential risks associated with automation in experimental design and reporting. Primary risks include (1) over-reliance on AI suggestions that might introduce unnoticed errors; (2) inadvertent release of sensitive data while preparing reproducible artifacts; and (3) misuse of benchmarking results to claim broader generalization than supported.

Mitigations: all AI-generated code and text were reviewed and validated by the human author; datasets used are public benchmarks (no private or sensitive datasets were used in the reported experiments); the paper explicitly documents limitations and avoids overstating generalization. Intended use: research benchmarking and tool-building; not for clinical or safety-critical decision making without further validation. Safeguards: code release includes instructions, seed settings, and randomized checks to reduce irreproducible behavior.

Reproducibility Statement

We release code, data preprocessing scripts, and raw result logs alongside the submission. Experiments were run using scikit-learn 1.x, Python 3.10, and standard CPU workers (Google Colab standard runtime). Random seeds were fixed where applicable; hyperparameters and train/test splits are included in the repository. Exact commands to reproduce the main tables and figures are provided in the supplementary material.

AI Involvement Checklist

Hypothesis development: Human-generated. The hypothesis and research goals were designed by the human author.

Experimental design and implementation: Mostly human, assisted by AI. AI suggested implementation strategies (e.g., safe aggregation logic and code snippets) but final design and execution were decided and run by the human author.

Analysis of data and interpretation of results: Human-generated. Data preprocessing, model training, and interpretation were performed by the human author.

Writing: Mostly human, assisted by AI. AI provided draft text and LaTeX boilerplate; the human author revised, reorganized, and finalized the full manuscript.

Editing / polishing: Mostly human, assisted by AI. AI suggested rewordings and formatting corrections; final edits were made by the human author.

Figures / tables: Mostly human, assisted by AI. AI helped generate LaTeX table code and caption text; plots and figure selection were performed or verified by the human author.

Paper Checklist

Does the paper clearly state its contributions? Yes. Section 1 lists the main contributions (multi-agent architecture, safe aggregation, evaluation, visualizations).

Are the limitations of this work discussed? Yes. The Limitations section describes scope limits (classical ML only, dataset sizes) and potential under-reporting caveats.

Is the work reproducible (code, data, pipeline released)? Yes. The manuscript states code and artifacts are released; the Reproducibility Statement indicates seeds, environment, and instructions.

Does the paper properly cite related work? Yes. Key benchmarking frameworks and dataset descriptions are cited.

Are AI contributions acknowledged transparently? Yes. The "AI and Human Contributions Statement" lists roles for both human and AI contributions and states that human oversight validated AI outputs.

References

[1] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[2] S. Aeberhard, D. Coomans, and O. De Vel, "Comparative analysis of statistical pattern recognition methods in high dimensional settings," Technical Report, Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland, 1992.

[3] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," *Biomedical Image Processing and Biomedical Visualization*, 1993.

[4] J. Vanschoren et al., "OpenML: networked science in machine learning," *ACM SIGKDD Explorations*, 2013.

[5] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems*, 2021.