

Identification Of Algorithm from the Given Dataset Using AI/ML Techniques

Miss. Sakshi D. Mahajan¹, Asst. Prof. Chetana M. Kawale²

Student, Department of Computer Applications, SSBT's College of Engineering & Technology Bambhori, Jalgaon.MH,
India¹

Guide, Department of Computer Applications, SSBT's College of Engineering & Technology Bambhori, Jalgaon.MH,
India²

Abstract: Selecting a suitable machine learning algorithm for a given dataset remains a persistent challenge due to the wide range of available algorithms and the varying statistical characteristics of real-world data. This paper presents a meta-learning based approach that recommends a suitable algorithm family for a given tabular dataset by analyzing its statistical meta-features, such as feature correlation, class imbalance, skewness, and dimensionality. Five candidate algorithms - Linear/Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest Neighbors - were evaluated on sixteen benchmark datasets (five real-world datasets and eleven synthetically generated datasets covering diverse characteristics such as class imbalance, noise, and high dimensionality), and a Random Forest meta-classifier was trained to map dataset meta-features to the best-performing algorithm. Using Leave-One-Out cross-validation, the system achieved a Top-1 recommendation accuracy of 31.25% and a Top-2 accuracy of 50.00%, compared to a naive baseline of 37.50%. While the small meta-dataset size limits definitive conclusions, the results and their limitations offer useful insight into the data requirements of meta-learning based algorithm recommendation systems, and feature importance analysis identified feature correlation and class imbalance as the strongest predictors of algorithm performance.

Keywords: Meta-learning, Algorithm Selection, Machine Learning, AutoML, Meta-features, Random Forest, Baseline Comparison

1. INTRODUCTION

Machine learning has become integral to data-driven decision-making across domains such as finance, healthcare, retail, and governance. However, the growing number of available algorithms, each suited to different types of data distributions and problem structures, makes algorithm selection a non-trivial task. A model that performs well on one dataset may perform poorly on another due to differences in feature correlation, class balance, dimensionality, or noise level. In current practice, practitioners address this by repeatedly training and evaluating multiple algorithms until an acceptable one is found, which is time-consuming and demands considerable expertise, particularly for those newer to machine learning. This research is motivated by the observation that datasets with similar statistical characteristics tend to respond well to similar categories of algorithms, a principle known as meta-learning. Rather than testing every algorithm exhaustively, a meta-learning system studies patterns from previously analyzed datasets and uses this knowledge to recommend a suitable algorithm for a new, unseen dataset. Such systems form the conceptual basis of modern Automated Machine Learning (AutoML) frameworks, which aim to reduce the manual burden of the model development pipeline.

The scope of this study is limited to structured tabular datasets used for supervised learning tasks, specifically classification and regression. The system recommends one of five widely used algorithm families: Linear/Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Unlike many published meta-learning studies that rely on hundreds of benchmark datasets, this work uses a modest set of sixteen datasets, allowing the study to additionally examine, as a secondary contribution, how meta-classifier performance is affected when the meta-dataset itself is small, a practical constraint often faced by student and early-stage researchers with limited computational resources and time.

The primary objectives of this research are: (1) to design and implement a meta-feature extraction and algorithm evaluation pipeline; (2) to train a meta-classifier that recommends a suitable algorithm for a new dataset; and (3) to critically evaluate this recommendation system against a naive baseline, in order to draw honest conclusions about its practical reliability at a small meta-dataset scale.

2. LITERATURE SURVEY

The algorithm selection problem was first formally described by Rice [1], who proposed a general framework for mapping problem characteristics to algorithm performance. This framework laid the theoretical foundation for meta-learning, where measurable dataset properties, referred to as meta-features, are used to predict which algorithm is best suited to a given problem.

Meta-learning approaches typically extract meta-features spanning simple statistics (number of instances and features), statistical measures (skewness, kurtosis, feature correlation), and information-theoretic measures (class entropy, imbalance ratio). Prior studies have shown that datasets with similar meta-feature profiles tend to favor similar algorithms, supporting the feasibility of recommendation systems built on these properties rather than exhaustive trial-and-error testing.

Automated Machine Learning (AutoML) tools such as Auto-WEKA, Auto-sklearn, and TPOT have demonstrated automated search across algorithms and hyperparameters using Bayesian optimization, genetic programming, or grid search, often combined with meta-learning warm-starts that use knowledge from previously seen datasets to narrow the search space [2]. While effective, these systems are computationally expensive, motivating lighter-weight meta-feature-based recommendation approaches such as the one proposed in this paper.

Comparative studies of classical algorithms, such as Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbors, and Logistic/Linear Regression, consistently find that no single algorithm performs best across all datasets, a result aligned with the No Free Lunch theorem [3], which states that no algorithm is universally superior across all possible problems. This finding directly motivates the need for a dataset-aware recommendation approach.

A recurring theme in the meta-learning literature, though less frequently emphasized, is the sensitivity of meta-classifier performance to the number of datasets used for meta-training. Most published meta-learning studies rely on repositories containing several hundred datasets to achieve reliable meta-model performance; studies using very small meta-dataset collections are less common in the literature, representing a gap this paper's secondary contribution addresses by explicitly reporting outcomes at this smaller scale.

3. RESEARCH METHODOLOGY

A. Data Collection

Sixteen benchmark datasets were used in this study: five real-world datasets (Iris, Wine, Breast Cancer, Digits, and Diabetes, sourced from the scikit-learn library) and eleven synthetically generated datasets created using scikit-learn's `make_classification` and `make_regression` utilities. The synthetic datasets were deliberately constructed to vary in class balance, noise level, dimensionality, and sample size, in order to diversify the meta-feature space used to train the recommendation system.

B. Meta-Feature Extraction

For each dataset, meta-features were computed across four categories: simple statistics (number of instances, number of features, ratio of instances to features), statistical measures (mean absolute feature correlation, average skewness, average kurtosis), information-theoretic measures (class imbalance ratio), and data quality measures (missing value ratio).

C. Candidate Algorithms

Five candidate algorithms were evaluated: Linear/Logistic Regression, a simple parametric baseline; Decision Tree, a non-parametric model using axis-aligned splits; Random Forest, an ensemble of decision trees; Support Vector Machine (SVM), which finds an optimal separating boundary; and K-Nearest Neighbors (KNN), an instance-based method.

D. Candidate Algorithm Evaluation

Each dataset was split into training and testing sets (80:20). All five candidate algorithms were trained and evaluated using accuracy and F1-score for classification, and RMSE and R^2 for regression. The best-performing algorithm on each dataset was recorded as the ground-truth label.

E. Baseline Comparison

To contextualize algorithm performance, a naive baseline was computed for each dataset: for classification, always predicting the majority class; for regression, always predicting the mean of the training target. This baseline represents the performance achievable with no learning at all.

F. Meta-Model Training and Evaluation

The resulting meta-dataset was used to train a Random Forest meta-classifier (200 estimators). Given the small size of the meta-dataset ($n=16$), Leave-One-Out cross-validation was used. Performance was measured using Top-1 and Top-2 accuracy, compared against a naive meta-baseline.

4. PROPOSED SYSTEM DESIGN

Figure 1 describes the proposed system architecture for algorithm recommendation from a given dataset. The system accepts a tabular dataset as input, which is first cleaned and normalized. The Meta-Feature Extraction module then computes the eight statistical meta-features described in Section 3-B, which are passed to the Candidate Algorithm Evaluation module during training, and to the trained Meta-Classifer during inference, which outputs a ranked list of recommended algorithms.



Figure 1: Proposed system architecture

Data Collection:

This module collects the sixteen benchmark datasets used in this study, comprising five real-world datasets and eleven synthetically generated datasets with deliberately varied statistical characteristics.

Meta-Feature Extraction and Normalization:

Each dataset is standardized using z-score normalization, after which the eight meta-features are computed and assembled into a fixed-length feature vector.

Algorithm Recommendation Generation:

The Meta-Classifer, a Random Forest trained on the meta-feature vectors and their best-performing algorithm labels, outputs a ranked list of recommended algorithms, filtered to the top two candidates for practical use.

5. RESULTS AND DISCUSSIONS

A. Candidate Algorithm Performance

Table 1 and Table 2 summarize the best-performing algorithm and score for each of the ten classification and six regression datasets respectively, alongside the naive baseline score for comparison.

Table 1: Classification Dataset Results (Best Algorithm vs. Baseline)

Dataset	Source	Best Algorithm	Accuracy	Baseline
Iris	Real	Logistic Reg.	1.0000	0.3000
Wine	Real	Logistic Reg.	1.0000	0.3889
BreastCancer	Real	Logistic Reg.	0.9737	0.6228
Digits	Real	SVM	0.9806	0.0778
SynClf_Balanced	Synthetic	SVM	0.9300	0.5100
SynClf_Imbalanced	Synthetic	KNN	0.9500	0.8200
SynClf_Noisy	Synthetic	Random Forest	0.7900	0.5000
SynClf_HighDim	Synthetic	Logistic Reg.	0.8000	0.4167
SynClf_MultiClass	Synthetic	SVM	0.7917	0.2083
SynClf_LowSample	Synthetic	Random Forest	0.8333	0.5417

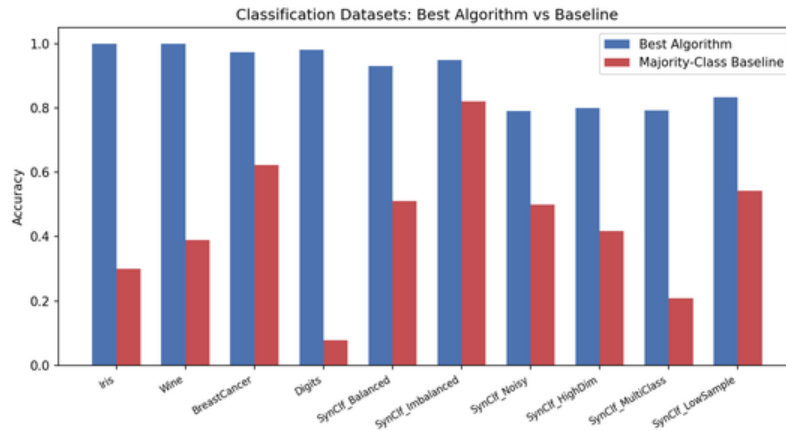


Figure 2 : Best algorithm vs. majority-class baseline across classification datasets.

On every classification dataset, the best candidate algorithm clearly outperformed the naive majority-class baseline, confirming that genuine learning was taking place rather than the models exploiting class imbalance. The performance margin was largest on Digits (0.9806 vs. 0.0778), which has ten balanced classes, and smallest on SynClf_Imbalanced, where the 85:15 class split gave the baseline an artificially high starting point.

Table 2 : Regression Dataset Results (Best Algorithm vs. Baseline)

Dataset	Source	Best Algorithm	R ² Score	Baseline R ²
Diabetes	Real	Linear Reg.	0.4526	-0.0120
SynReg_LowNoise	Synthetic	Linear Reg.	0.9984	-0.0363
SynReg_HighNoise	Synthetic	Linear Reg.	0.9290	-0.0268
SynReg_HighDim	Synthetic	Linear Reg.	0.9964	-0.0004
SynReg_SparseInformative	Synthetic	Linear Reg.	0.8957	-0.0150
SynReg_LowSample	Synthetic	Linear Reg.	0.9942	-0.0138

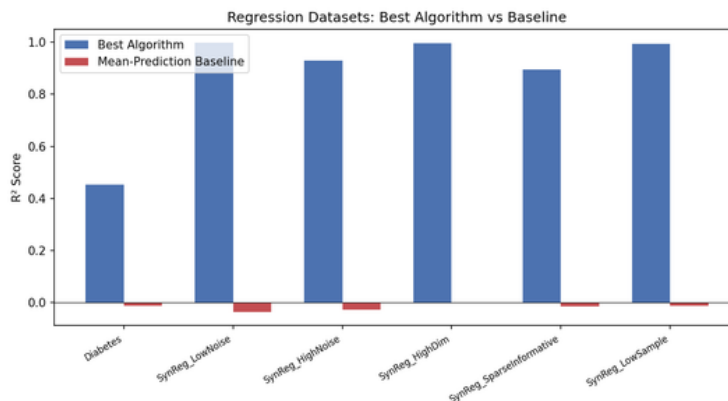


Figure 3: Best algorithm vs. mean-prediction baseline across regression datasets.

Linear Regression was the best-performing algorithm on all six regression datasets, including the real-world Diabetes dataset and all five synthetic regression variants. This is expected, since the synthetic regression datasets were generated using an underlying linear function, and even the real Diabetes dataset is known in the literature to have a largely linear relationship between its clinical features and disease progression.

B. Meta-Classifer Performance and Comparison to Baseline

The meta-classifier, trained on the meta-features of all sixteen datasets and evaluated using Leave-One-Out cross-validation, achieved a Top-1 accuracy of 31.25% (5 out of 16) and a Top-2 accuracy of 50.00% (8 out of 16). For comparison, a naive meta-baseline that always recommends the single most frequent best-algorithm label in the meta-dataset (Linear Regression, which was the best performer on 6 of 16 datasets) achieved 37.50% (6 out of 16). Figure 4 presents this comparison.

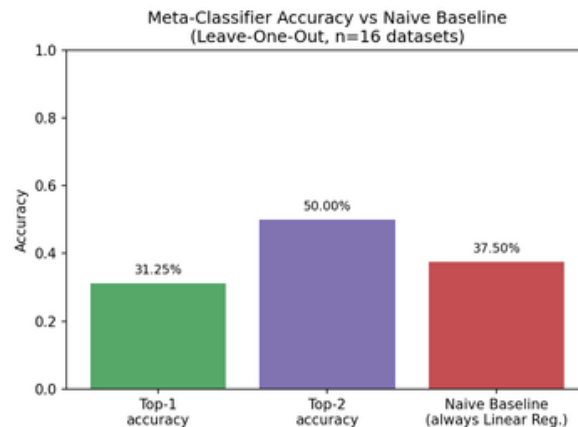


Figure 4: Meta-classifier Top-1 and Top-2 accuracy compared to a naive baseline (Leave-One-Out, n=16).

Notably, the meta-classifier's Top-1 accuracy (31.25%) was lower than the naive baseline (37.50%). This is an important and honest finding rather than a favorable one: with only sixteen datasets and eight meta-features, the meta-classifier had insufficient data to learn a reliable mapping and, evaluated one dataset at a time via Leave-One-Out, was prone to overfitting on the remaining fifteen. Top-2 accuracy did exceed the baseline, suggesting the model retained some genuine signal, but the results as a whole indicate that sixteen datasets are not sufficient to train a meta-classifier that reliably outperforms a simple heuristic. This is consistent with the broader meta-learning literature, where published systems typically rely on meta-training collections of several hundred datasets; the present study's smaller scale should therefore be read as a controlled demonstration of pipeline feasibility and of the data requirements of meta-learning, rather than as evidence of a production-ready recommendation tool.

C. Feature Importance Analysis

Figure 5 shows the average feature importance, across all Leave-One-Out folds, assigned by the Random Forest meta-classifier to each meta-feature.

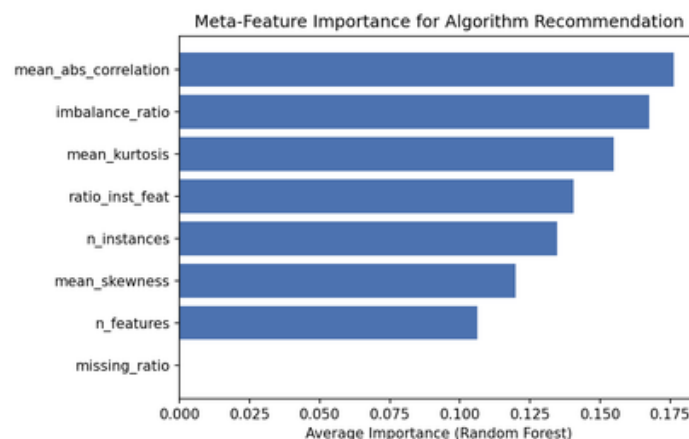


Figure 5: Average meta-feature importance across Leave-One-Out folds.

Mean absolute feature correlation and class imbalance ratio were the two most influential meta-features, together accounting for over a third of the total importance. This aligns with the intuition that highly correlated features tend to favor linear models, while class imbalance disproportionately affects distance-based and probabilistic classifiers such as KNN and Logistic Regression. The missing-value ratio meta-feature carried zero importance, which is expected since none of the sixteen datasets used in this study contained missing values.

D. Limitations

The central limitation of this study is the size of the meta-dataset. Sixteen datasets, while sufficient to demonstrate the end-to-end meta-learning pipeline, are not enough to train a meta-classifier that reliably beats a naive baseline, as the results in Section 5-B show. Additionally, eleven of the sixteen datasets are synthetically generated; while this allowed deliberate control over dataset characteristics such as noise and imbalance, it does not fully substitute for the variety found in real-world data. The candidate algorithm pool was also limited to five classical algorithms, excluding gradient boosting methods and neural networks that are increasingly used in practice.

6. CONCLUSION & FUTURE WORK

This paper implemented and evaluated a complete meta-learning pipeline for recommending machine learning algorithms based on dataset meta-features. Candidate algorithm evaluation on sixteen benchmark datasets confirmed that no single algorithm dominates across all datasets, and that learned models substantially outperform naive baselines. However, the meta-classifier itself, trained on only sixteen datasets, achieved a Top-1 accuracy (31.25%) below that of a naive baseline (37.50%), and a Top-2 accuracy (50.00%) only moderately above it. This honest result highlights that meta-dataset scale, rather than meta-model sophistication, is likely the primary bottleneck for this type of system, and that studies of this kind require substantially more than sixteen datasets to produce a meta-classifier that is practically reliable.

Future Work: Future work will focus on expanding the meta-dataset to at least fifty to one hundred datasets, incorporating additional candidate algorithms such as gradient boosting methods, and exploring simpler, more data-efficient meta-models such as k-nearest-neighbor-based meta-learners. A web-based interface is also planned to make the tool accessible without requiring coding knowledge.

REFERENCES

- [1]. J. R. Rice, "The algorithm selection problem," *Advances in Computers*, vol. 15, pp. 65-118, 1976.
- [2]. M. Feurer, A. Klein, K. Eggenberger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [3]. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.
- [4]. R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77-95, 2002.
- [5]. J. Smith, *Introduction to Machine Learning*. Springer, 2020.
- [6]. UCI Machine Learning Repository. [Online]. Available: <https://archive.ics.uci.edu/>
- [7]. Scikit-learn Developers. Scikit-learn: Machine Learning in Python. [Online]. Available: <https://scikit-learn.org/>

APPENDIX

The Python implementation used to generate all results reported in this paper is available as a supplementary code file (`meta_learning_system.py`) accompanying this submission, and is fully reproducible given the fixed random seeds used throughout.