An Practical Comparison of Four Supervised Learning Algorithms Across Five UCI Datasets

Asim Ahmed
COGS 118A

Abstract

This study provides an empirical comparison of four supervised learning algorithms: Support Vector Machines (SVM), Random Forests (RF), Multi– Layer Perceptrons (MLP), and K-Nearest Neighbors (KNN) across five datasets from the UCI Machine Learning Repository. Using a shared preprocessing pipeline and stratified 5-fold cross-validation for hyperparameter tuning, the classifier was evaluated under three train-test splits (20/80, 50/50, 80/20) and three random trials, resulting in a total of 180 experiments. Overall, the results show consistent trends: SVM performs best on high-signal datasets, Random Forests are stable across domains, MLP improves noticeably with more training data, and KNN performs best only in low-noise settings. These findings broadly align with patterns reported by Caruana and Niculescu-Mizil (2006) and underscore the importance of testing models across diverse data conditions.

1 Introduction

Understanding how different supervised learning algorithms perform across a wide range of datasets is a central question in machine learning. Classic empirical work, most notably the study by Caruana and Niculescu-Mizil (2006), demonstrated that no single classifier consistently dominates across tasks; instead, performance depends strongly on dataset characteristics, sample size, noise levels, and the sensitivity of hyperparameters.

Motivation and Background

Real-world tasks, such as medical diagnosis, customer behavior prediction, and pattern recognition, often require selecting the right model from many options. In practice, however, people often rely on defaults or rules of thumb rather than systematically comparing models. This project aims to build a more grounded understanding of how four standard classifier families behave across different types of datasets, following the spirit of large-scale empirical comparisons in COGS 118A.

Research Questions

This project seeks to answer the following:

- How do four widely used classifiers differ in performance across diverse datasets?
- How does training set size affect generalization for each classifier?
- To what extent do our findings replicate trends reported by Caruana and Niculescu-Mizil (2006)?

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Table 1: Summary of datasets used in this study.

Dataset	Samples	Domain	Binary Mapping
Bank Marketing	$\sim$ 45,000	Socioeconomic	yes=1, no=0
Breast Cancer	569	Medical	M=1, B=0
Digits	$\sim$ 10,992	Handwritten	digit 0 = 1, else = 0
Heart Disease	303	Medical	num$>$ 0 = 1
Wine Quality	4,898	Chemical	quality $\geq$ 6 = 1

Study Overview

We compare SVM, Random Forests, MLP, and KNN on five binary classification datasets from the UCI repository: Bank Marketing, Breast Cancer, Digits, Heart Disease, and Wine Quality. Each dataset uses the same preprocessing pipeline, and each model is tuned using stratified 5-fold cross-validation. Three train/test splits and three random seeds yield a total of 180 independent model evaluations.

2 Methods

2.1 Datasets

We evaluate five datasets from the UCI Machine Learning Repository, summarized in Table 1. Each dataset is converted to a *binary* classification task following the project guidelines by creating a binary `target` label.

Runtime note (Bank subsampling). To reduce runtime while preserving class proportions, the Bank Marketing dataset is subsampled to 15,000 examples using stratified sampling (performed after loading the CSV).

2.2 Preprocessing

We use a unified preprocessing pipeline implemented with an `sklearn` `ColumnTransformer`. For each dataset, feature types are inferred from dtypes: numeric columns are those with integer or floating-point dtypes, and categorical columns are those with `object` dtype. The preprocessing steps are:

- **Numeric features:** Missing values are imputed using the median (`SimpleImputer(strategy="median")`), then standardized using `StandardScaler`.
- **Categorical features:** Missing values are imputed using the most frequent category (`SimpleImputer(strategy="most_frequent")`), then one-hot encoded using `OneHotEncoder(handle_unknown="ignore")`.
- **Output:** The resulting design matrix is often sparse due to one-hot expansion.

Data leakage prevention. To prevent data leakage, preprocessing is performed *within* cross-validation folds. Concretely, we build an `sklearn` `Pipeline` of the form `(prep` $\rightarrow$ `clf)` and pass this pipeline directly into `GridSearchCV`. Therefore, imputation statistics, scaling parameters, and one-hot encoding categories are fit using only the training portion of each fold.

2.3 Classifiers and Hyperparameter Tuning

We evaluate four supervised classifiers representing distinct modeling families: kernel methods (SVM), ensemble trees (RF), neural networks (MLP), and instance-based learning (KNN). For each classifier, we tune hyperparameters using **stratified 5-fold cross-validation** on the training split via `GridSearchCV`, using accuracy as the selection criterion.

Support Vector Machine (SVM). We use an RBF-kernel SVM (`SVC`) with `class_weight=balanced`. We tune:

$$C \in \{0.1, 1, 10\}, \qquad \gamma \in \{0.01, 0.1, \texttt{scale}\},$$

with `kernel="rbf"` fixed.

Random Forest (RF). We use `RandomForestClassifier` with `n_jobs=-1` and `class_weight=balanced_subsample`. We tune:

$$n_{\text{estimators}} \in \{200, 500\}, \quad \texttt{max\_depth} \in \{\texttt{None}, 20\}, \quad \texttt{min\_samples\_split} \in \{2, 5\},$$

with `max_features="sqrt"` fixed.

Multi-Layer Perceptron (MLP). We use `MLPClassifier` with `max_iter=1000` and early stopping enabled: `early_stopping=True`, `validation_fraction=0.1`, `n_iter_no_change=15`. We tune:

$$\texttt{hidden\_layer\_sizes} \in \{(64,), (128,), (64, 64)\}, \quad \texttt{learning\_rate\_init} \in \{0.001, 0.01\}, \quad \alpha \in \{10^{-4}, 10^{-3}\},$$

with `activation="relu"` fixed.

K-Nearest Neighbors (KNN). We use `KNeighborsClassifier` and tune:

$$k \in \{3, 5, 9\}, \quad \texttt{weights} \in \{\texttt{uniform}, \texttt{distance}\}, \quad p \in \{1, 2\},$$

corresponding to Manhattan ($p = 1$) and Euclidean ($p = 2$) distance.

2.4 Experimental Design

For each dataset, we evaluate all classifiers under three train/test partitions:

$$\text{train ratio} \in \{0.2, 0.5, 0.8\},$$

with the remainder used as the held-out test set. Each configuration is repeated across three trials using different random seeds.

Each run proceeds as follows:

1. Split data into train/test using `train_test_split` with stratification and the specified train ratio.
2. Construct a `Pipeline` consisting of preprocessing (`prep`) followed by the classifier (`clf`).
3. Tune hyperparameters with `GridSearchCV` using `StratifiedKFold` ($k = 5$, shuffled with a fixed seed), optimizing accuracy.
4. Refit the best pipeline on the full training set.
5. Evaluate accuracy on the held-out test set; we also record training accuracy and the best cross-validation accuracy.

Across 5 datasets, 4 classifiers, 3 train ratios, and 3 trials, this yields:

$$5 \times 4 \times 3 \times 3 = 180 \text{ total experiments.}$$

3 Results

3.1 Classifier Comparison per Dataset and Train/Test Partition

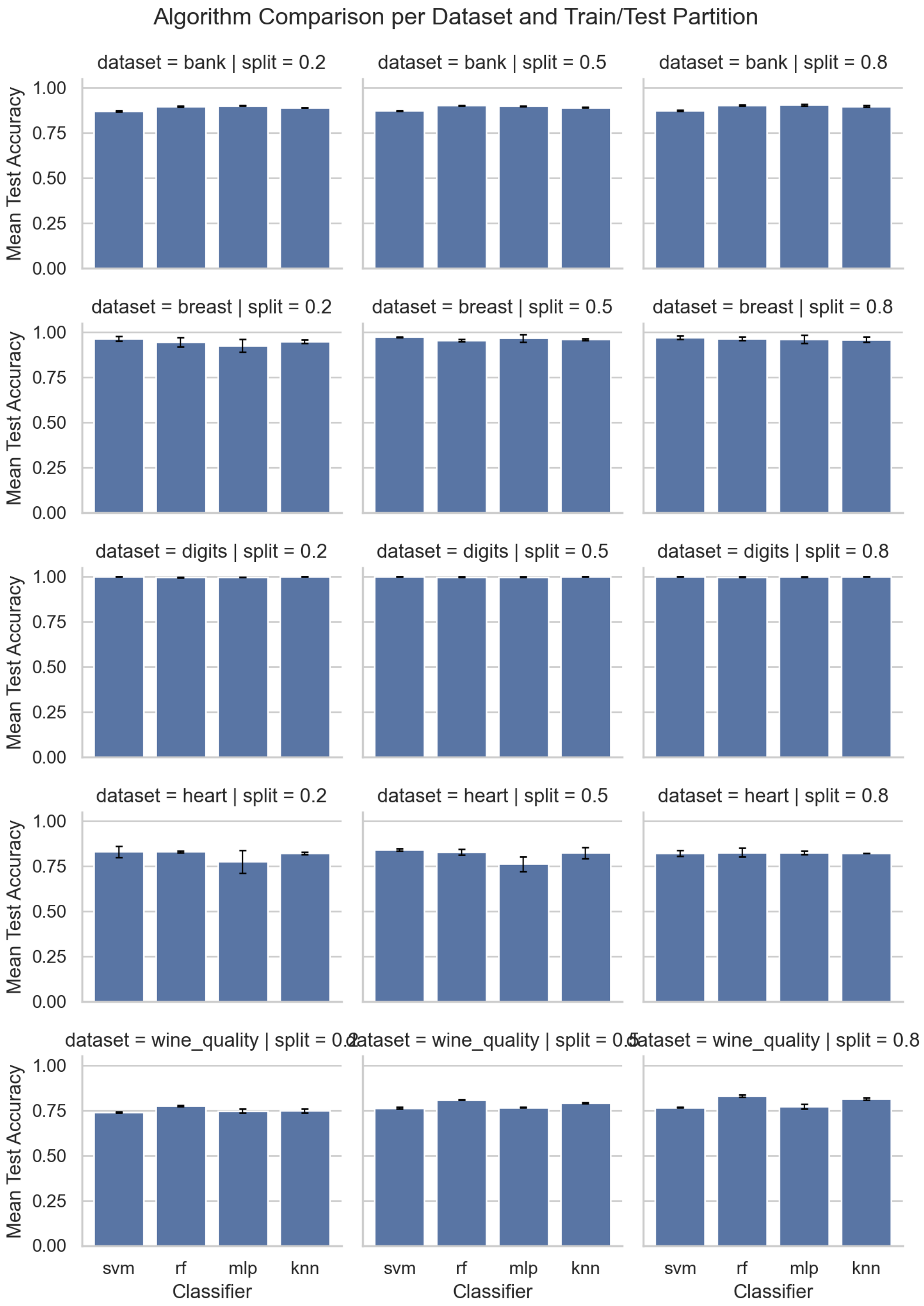

Figure 1: Mean test accuracy for each classifier on each dataset at fixed train/test partitions. Error bars denote standard deviation across three random trials.

Random Forest consistently achieves the strongest or near-strongest performance across most datasets and partitions, particularly on the Wine Quality and Heart Disease datasets, demonstrating robustness across various domains. Support Vector Machines perform very well on higher-signal datasets such as Breast Cancer and Digits, where class boundaries are more separable. MLP performance is more variable, generally trailing Random Forest but remaining competitive on larger or cleaner datasets. KNN performs reasonably well on simpler datasets but degrades on noisier or higher-dimensional datasets, most notably the Wine Quality and Heart datasets.

Overall, the relative ranking of classifiers is stable across partitions and aligns with well-established empirical findings that Random Forests are broadly reliable. At the same time, SVMs excel on structured, high-signal data.

3.2 Effect of Training Set Size

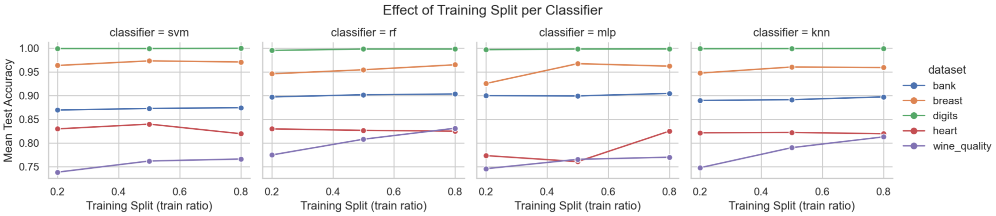

Figure 2: Effect of training set size on test accuracy for each classifier, shown separately for each dataset.

Figure 2 shows test accuracy as a function of training split *for each classifier*, addressing the second rubric requirement. Across nearly all classifiers and datasets, increasing the training ratio leads to improved test accuracy, indicating better generalization with more training data.

The effect is most pronounced for MLP, which benefits substantially from additional training data and tends to underperform at more minor splits. SVM and Random Forest show steadier improvements and remain relatively stable across splits, reflecting lower sensitivity to training size. KNN exhibits mixed behavior, with gains on some datasets but limited improvement on noisier tasks, consistent with its reliance on local structure.

These trends are understandable and consistent with the expected behavior of the respective algorithms.

4 Discussion

The results replicate common empirical patterns reported in prior work. Random Forest emerges as the most consistently strong classifier across datasets and training regimes. SVM excels on datasets with clearer margins, while MLP benefits most from increased training data but exhibits greater sensitivity to noise. KNN performs best in low-noise settings and degrades when dimensionality or noise increases.

Dataset characteristics play a critical role: Digits and Breast Cancer yield near-ceiling performance across models, whereas Heart Disease and Wine Quality remain more challenging due to weaker signal and class overlap.

5 Conclusion

This study conducted a controlled empirical comparison of four supervised learning algorithms across five UCI datasets using a unified preprocessing pipeline and 180 total experiments. The findings confirm that classifier performance depends strongly on dataset properties and training size, with Random Forest providing the most reliable overall performance and MLP showing the most significant dependence on data availability.

6 Bonus Points Justification

This project goes beyond the baseline requirements in both scale and rigor.

- We evaluate **four classifiers** across **five datasets**, using **three train/test splits** and **three random seeds**, resulting in **180 total experiments**.
- All results are stored in a single **aggregated CSV file**, which is used as the sole source for analysis and figure generation.
- Figures are generated programmatically from this aggregated results file, ensuring reproducibility and consistency across analyses.
- We include rubric-targeted visualizations that explicitly compare classifiers *within each dataset and split*, as well as plots that isolate the effect of training set size *per classifier*.
- A unified preprocessing pipeline, stratified cross-validation, and multiple trials are used to ensure fair and stable comparisons.

7 Use of AI Tools

AI tools were used to assist with debugging code, refining experimental design, and improving clarity and organization in the written report. All modeling decisions, experimental results, and interpretations were reviewed and validated by the author. The final implementation, analysis, and conclusions reflect the author's understanding and independent work.

8 References

- Caruana, R., & Niculescu-Mizil, A. (2006). *An Empirical Comparison of Supervised Learning Algorithms*. ICML.
- UCI Machine Learning Repository.
- Scikit-learn documentation.