Empirical Comparison of Linear, Ensemble, and Non-Linear models on Diverse Classification Tasks

Jiaqi Wu [1]

Abstract

This paper presents a comprehensive empirical comparison of three classification algorithms—Logistic Regression, Random Forests, and Neural Networks—on three diverse binary classification datasets from the UCI Machine Learning Repository. Following the methodology of Caruana and Niculescu-Mizil (2006), we conduct a total of 81 experiments with hyperparameter tuning via 5-fold cross-validation. Results demonstrate that Logistic Regression provides a reliable, interpretable baseline with consistent performance and minimal overfitting, while Random Forests achieve the highest average accuracy across most configurations and show the largest gains from increased training data. All classifiers benefit from increased training data, with performance trends aligning with Caruana and Niculescu-Mizil's findings.

1. Introduction

Classification is a fundamental task in machine learning with applications across diverse domains. This paper presents an empirical comparison of three widely-used classification algorithms, replicating and extending the influential study by Caruana and Niculescu-Mizil (2006) (Caruana & Niculescu-Mizil, 2006). We evaluate three classifiers on binary classification tasks: Logistic Regression, a linear probabilistic classifier with interpretable coefficients; Random Forests, an ensemble method combining multiple decision trees with bagging; and Neural Networks, specifically multi-layer perceptrons (MLPs) trained via backpropagation. These classifiers represent different paradigms—linear models, ensemble learning, and deep learning—providing a comprehensive comparison across the complexity spectrum.

1.1. Experimental Scope

We conduct a rigorous empirical study evaluating three classifiers from different families on three UCI datasets for binary classification. Our experimental design includes three train-test partitions (20/80, 50/50, 80/20) with three independent trials per configuration for statistical reliability. Hyperparameter tuning is performed using 5-fold cross-validation, resulting in a total of 81 experiments (3 trials × 3 classifiers × 3 datasets × 3 partitions). For each experiment, we report three metrics: training accuracy, validation accuracy from cross-validation, and test accuracy, enabling comprehensive analysis of overfitting and generalization.

2. Datasets

2.1. Dataset Selection and Characteristics

We selected three diverse datasets from the UCI Machine Learning Repository that represent different domains, scales, and classification challenges. This diversity ensures our findings generalize across various real-world scenarios.

The **BEED** (Bangalore EEG Epilepsy Dataset) (. & BANU P K, 2024) contains 8000 subjects with 16 continuous features derived from EEG signal characteristics. This dataset is designed for binary classification between epileptic and non-epileptic subjects, providing a medical signal processing application with balanced class distribution. The features capture various spectral and temporal properties of EEG recordings.

The **Student Dropout and Academic Success** dataset (Realinho & Baptista, 2021) contains 4424 instances with 36 features encompassing demographic, academic, and socioeconomic variables. For binary classification, we merged multiple outcome labels (Enrolled and Graduate) into a Non-Dropout class competing against Dropout, resulting in a moderately imbalanced distribution (approximately 35% Dropout, 65% Non-Dropout).

The **Poker Hand** dataset (Cattral & Oppacher, 2002) originally contains 1,025,010 instances with 10 integer features representing five playing cards (suit and rank for each card). Due to computational constraints, we sampled a subset of 10,000 instances to maintain class distribution. For binary classification, we map the target to distinguish between no-pair hands (class 0) and any winning hand (classes 1-9), resulting in a nearly balanced distribution. This dataset tests classifier performance on pattern recognition tasks with integer features and provides insights into classifier behavior

on gaming domain applications.

2.2. Data Preprocessing

We applied consistent preprocessing to ensure fair comparison across all classifiers. For the Poker Hand dataset, we first sampled 10,000 instances using stratified sampling to maintain class distribution before any preprocessing steps. All three datasets contain only numeric features (no categorical features requiring encoding). For datasets with multi-class labels, we converted them to binary classification using dataset-specific strategies applied during the experiment loop: The BEED dataset originally contains four classes (0, 1, 2, 3); we collapsed these into binary by grouping classes 0 and 1 as negative (non-epileptic) and classes 2 and 3 as positive (epileptic). The Student dataset contains multiple outcome labels (Dropout, Enrolled, Graduate); we merged Enrolled and Graduate into a Non-Dropout class competing against Dropout. The Poker Hand dataset contains ten classes representing different hand types (0: no pair, 1-9: various winning hands); we mapped class 0 as negative (no-pair) and classes 1-9 as positive (any winning hand). No missing values were found in the original datasets. Feature scaling using standardization (zero mean, unit variance): $x' = \frac{x-\mu}{\sigma}$ was applied after train-test splitting, fitted only on training data and then applied to validation/test sets to prevent data leakage. Scaling was used only for Logistic Regression and Neural Networks; Random Forests used unscaled features.

3. Methods

3.1. Logistic Regression

Logistic Regression models (Müller, 2004) the probability of class membership using the logistic function:

$$P(y = 1|x) = \frac{1}{1 + e^{-w^T x}}$$

where w represents the weight vector and x is the feature vector. The model is trained by maximizing the log-likelihood of the observed data, which is equivalent to minimizing the cross-entropy loss function. This optimization problem is convex, ensuring convergence to a global optimum. L1 and L2 regularization are used to prevent overfitting, controlled by the inverse regularization parameter C where smaller values indicate stronger regularization. The choice of solver depends on the penalty type: liblinear supports both L1 and L2 penalties and is efficient for small to medium datasets, while lbfgs is limited to L2 penalty but scales better for larger problems. Logistic Regression provides interpretable coefficients that indicate the direction and magnitude of each feature's contribution to the classification decision, making it valuable for applications requiring model transparency.

3.2. Random Forests

Random Forests (Breiman, 2001) is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions. Key hyperparameters include the number of trees ($n_{\text{estimators}} \in \{50, 100, 200, 500\}$), maximum depth (max_depth $\in \{10, 20, 30, \text{None}\}$), minimum samples split (min_samples_split $\in \{2, 5, 10\}$), and maximum features (max_features $\in$ {'sqrt', 'log2'}). Random Forests handle non-linear relationships, are resistant to overfitting, and provide feature importance measures.

3.3. Neural Networks

Neural networks (Schmidhuber, 2015) serve as universal function approximators capable of capturing complex non-linear patterns. We use multi-layer perceptrons (MLPs) trained via backpropagation with the Adam optimizer. The architecture consists of an input layer, hidden layers, and an output layer with sigmoid activation. Key hyperparameters include hidden layer sizes ($\{(50,), (100,), (100, 50), (100, 100)\}$), learning rate ($\alpha \in \{0.0001, 0.001, 0.01\}$), ReLU activation function, maximum iterations (1000), and regularization strength (alpha $\in \{0.0001, 0.001, 0.01\}$).

4. Experiments and Results

This section presents a comprehensive empirical evaluation comparing Logistic Regression, Random Forests, and Neural Networks across three diverse binary classification datasets, systematically varying training set sizes to assess performance and generalization capabilities. Our experimental design follows a factorial structure with three train-test partitions (20/80, 50/50, 80/20) to examine the effect of training data availability, three independent trials per configuration using random seeds [42, 123, 456] for statistical reliability, and rigorous hyperparameter optimization via 5-fold cross-validation with grid search. This design yields a total of 81 experiments (3 trials × 3 classifiers × 3 datasets × 3 partitions), enabling robust statistical analysis and comprehensive performance characterization. For each experiment, we report three key metrics: training accuracy to assess model fit, validation accuracy from cross-validation to guide hyperparameter selection, and test accuracy to evaluate generalization, all reported as mean ± standard deviation across the three independent trials.

4.1. Algorithm Performance by Dataset

Tables 2, 3 ,and 4 show detailed results for BEED, Student Dropout, and Poker Hand datasets respectively, with training, validation, and test accuracies.

CLASSIFIER	20/80	50/50	80/20
LR	67.9 ± 16.1	68.2 ± 16.4	68.8 ± 16.1
RF	**80.3** ± 15.5	**82.9** ± 14.4	**83.6** ± 14.1
NN	79.0 ± 17.1	81.2 ± 17.4	81.5 ± 17.9
AVERAGE	75.7	77.4	78.0

Table 1. AVERAGE TEST ACCURACY (%) ACROSS DATASETS AND PARTITIONS. VALUES ARE MEAN ± STD OVER 3 TRIALS. BEST RESULT PER PARTITION SHOWN IN **BOLD**. ABBREVIATIONS: LR=LOGISTIC REGRESSION, RF=RANDOM FOREST, NN=NEURAL NETWORK.

PARTITION	CLASSIFIER	TRAIN	VAL	TEST
20/80	LR	67.75	66.67	66.60 ± 0.71
	RF	100.00	93.58	94.07 ± 0.52
	NN	97.25	94.67	94.81 ± 0.68
50/50	LR	66.65	66.64	66.53 ± 0.43
	RF	100.00	96.05	96.76 ± 0.20
	NN	99.43	97.01	97.68 ± 0.22
80/20	LR	67.38	67.23	68.02 ± 0.15
	RF	100.00	97.24	97.67 ± 0.21
	NN	99.78	97.96	98.77 ± 0.24

Table 2. BEED RESULTS: TRAIN/VAL/TEST ACCURACY (%) FOR EACH CLASSIFIER AND PARTITION. MEAN ± STD OVER 3 TRIALS.

The experimental results reveal distinct performance patterns across datasets that highlight the importance of matching algorithm characteristics to problem structure. On the BEED dataset, which requires non-linear decision boundaries for EEG signal classification, both Random Forests and Neural Networks achieve excellent performance, substantially outperforming Logistic Regression, with Random Forests demonstrating remarkable robustness through effective ensemble-based regularization. The Student Dropout dataset presents a different pattern, where Logistic Regression matches or exceeds non-linear methods, suggesting that demographic and academic features exhibit predominantly linear relationships well-captured by the simple linear model. The Poker Hand dataset reveals the most dramatic differences: Logistic Regression performs near chance level, demonstrating that intricate rule-based patterns cannot be captured by linear boundaries, while Random Forests significantly outperform Neural Networks, suggesting that tree-based ensembles' explicit feature interactions are particularly well-suited for this combinatorial pattern recognition problem. Overall, no single classifier dominates across all scenarios: Logistic Regression provides a stable, interpretable baseline with excellent generalization for linear problems, Random Forests achieve the highest average performance with robust generalization, making them an excellent default choice, and Neural Networks show competitive results but require more careful tuning to achieve optimal performance.

PARTITION	CLASSIFIER	TRAIN	VAL	TEST
20/80	LR	87.67	86.39	87.13 ± 0.41
	RF	97.25	86.46	86.61 ± 0.62
	NN	90.16	85.52	85.27 ± 1.24
50/50	LR	87.82	87.30	87.93 ± 0.39
	RF	98.37	86.99	87.43 ± 0.50
	NN	88.55	87.00	86.93 ± 0.44
80/20	LR	87.93	87.42	87.72 ± 0.13
	RF	96.58	87.45	87.31 ± 0.57
	NN	90.54	86.99	87.16 ± 0.43

Table 3. STUDENT DROPOUT RESULTS: TRAIN/VAL/TEST ACCURACY (%) FOR EACH CLASSIFIER AND PARTITION. MEAN ± STD OVER 3 TRIALS.

PARTITION	CLASSIFIER	TRAIN	VAL	TEST
20/80	LR	51.53	50.43	50.08 ± 0.39
	RF	98.85	59.88	60.10 ± 0.65
	NN	64.05	56.25	56.84 ± 0.08
50/50	LR	50.59	50.15	50.28 ± 0.28
	RF	95.84	63.54	64.40 ± 0.37
	NN	61.51	57.99	58.88 ± 1.09
80/20	LR	50.46	50.21	50.62 ± 0.89
	RF	95.05	65.09	65.70 ± 1.29
	NN	62.62	58.93	58.70 ± 1.49

Table 4. POKER HAND RESULTS: TRAIN/VAL/TEST ACCURACY (%) FOR EACH CLASSIFIER AND PARTITION. MEAN ± STD OVER 3 TRIALS.

4.2. Training Set Size Effect

CLASSIFIER	20/80	50/50	80/20	Δ(20→80)
LR	67.9	68.2	68.8	+0.9
RF	80.3	82.9	83.6	+3.3
NN	79.0	81.2	81.5	+2.5

Table 5. AVERAGE TEST ACCURACY IMPROVEMENT (%) WITH INCREASED TRAINING DATA. VALUES SHOW MEAN ACROSS ALL DATASETS.

The training set size analysis reveals distinct data efficiency patterns across classifiers. Random Forests show the largest improvement (+3.3%) when training data increases from 20% to 80%, benefiting from increased sample diversity for constructing diverse trees. Neural Networks exhibit moderate gains (+2.5%), indicating good data efficiency but with diminishing returns. Logistic Regression shows the smallest

improvement (+0.9%), reflecting its limited capacity to benefit from additional data once the linear decision boundary is well-estimated. All classifiers follow the expected trend of increasing test accuracy with more training data, validating the learning curve hypothesis and demonstrating that algorithm selection should be guided by data availability.

4.3. Hyperparameter Analysis

The hyperparameter analysis reveals consistent patterns across classifiers: Logistic Regression favors strong regularization $C = 0.1$ with L1 penalty. Random Forests consistently select moderate tree counts (200 estimators) with constrained depth, balancing bias and variance. Neural Networks favor deeper architectures (100, 100) with strong regularization ($\alpha = 0.0001$) and moderate learning rates ($\eta = 0.01$). These patterns reflect the different regularization needs of each algorithm class, with linear models requiring stronger regularization to prevent overfitting, ensemble methods benefiting from moderate complexity, and neural networks needing both architectural constraints and regularization penalties.

CLASSIFIER	HYPERPARAMETERS
LR	$C = 0.1$, PENALTY='L1', SOLVER='LIBLINEAR'
RF	$n_{\text{EST}} = 200$; $d_{\max} \in \{10, 30\}$; $s_{\min} = 2$; MAX_FEATURES='SQRT'
NN	LAYERS=$(100, 100)$; $\alpha = 0.0001$; $\eta = 0.01$

Table 6. MOST COMMON OPTIMAL HYPERPARAMETERS ACROSS ALL EXPERIMENTS.

5. Conclusion

Through rigorous empirical evaluation, this study replicates and validates the key findings of (Caruana & Niculescu-Mizil, 2006) on modern ML framework – scikit-learn. Our results confirm that algorithm selection depends on problem characteristics, data availability, and interpretability requirements. No single classifier dominates across all scenarios, emphasizing the importance of empirical evaluation for each specific application. Random Forests provide the best average performance and robustness, making them an excellent default choice, while Logistic Regression offers competitive performance on linearly separable problems with interpretability advantages, and Neural Networks require more careful tuning but show strong performance when properly regularized. The comprehensive experimental design provides reliable, reproducible results that practitioners can use to guide algorithm selection in binary classification tasks.

References

., N. and BANU P K, N. BEED: Bangalore EEG Epilepsy Dataset . UCI Machine Learning Repository, 2024. DOI: https://doi.org/10.24432/C5K33B.

Breiman, L. Random forests. *Mach. Learn.*, 45(1): 5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 161–168, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143865. URL https://doi.org/10.1145/1143844.1143865.

Cattral, R. and Oppacher, F. Poker Hand. UCI Machine Learning Repository, 2002. DOI: https://doi.org/10.24432/C5KW38.

Müller, M. Generalized linear models. 02 2004. doi: 10.1007/978-3-642-21551-3_24.

Realinho, Valentim, V. M. M. M. J. and Baptista, L. Predict Students' Dropout and Academic Success. UCI Machine Learning Repository, 2021. DOI: https://doi.org/10.24432/C5MC89.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015. ISSN 0893-6080. doi: 10.1016/j.neunet.2014.09.003. URL http://dx.doi.org/10.1016/j.neunet.2014.09.003.