Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 2, 2026

When Are Linear Classifiers Sufficient? A Re-evaluation and Conditional Boundary Exploration Based on OpenML

Lujuan Zhang

School of Computer Science and Artificial Intelligence (Industrial Software), Guangdong University of Science & Technology, Dongguan, Guangdong, China

Abstract: The point at which a linear classifier stops being "sufficient" lacks a quantitative definition. To address this, we evaluated 1,687 records from 298 OpenML datasets across three model families. Using a practical significance framework, we defined linear sufficiency as an accuracy drop within a tolerance threshold δ. At δ=0.02, linear sufficiency was observed in 68.9% of SVM datasets, 62.4% of NN datasets, but only 46.1% of DT datasets. Shallow decision tree extraction highlighted low feature-to-instance ratios and small class counts as primary predictors. However, leave-one-out validation exposed a clear split. The extracted rules worked well for the DT family (0.686 accuracy vs. 0.539 baseline) but failed to surpass baselines for SVM and NN. This suggests that while simple meta-features can predict linear sufficiency for decision trees, they cannot fully capture the complex boundaries of SVMs and neural networks.

Keywords: Linear Classifiers; Non-linear Classifiers; Algorithm Selection; Meta-learning; OpenML Benchmark.

1. Introduction

Choosing which model to deploy is rarely a trivial step in machine learning. Linear classifiers remain a common starting point because they train quickly, are easy to explain, and resist overfitting. The European Union's General Data Protection Regulation (GDPR) has reinforced this preference. With the "right to explanation" now legally codified [4], interpretability has shifted from a desirable feature to a hard business requirement. Yet a persistent rule of thumb still shapes model selection: non-linear models are presumed necessary for large datasets, while linear models are treated as a reasonable choice only when data is scarce or the problem is inherently linear.

Strang et al. (2018) [1] tested this assumption directly. Comparing linear and non-linear versions of SVMs, neural networks, and decision trees across 299 OpenML [2] datasets, they found that non-linear models often won—but linear models were not significantly worse on a surprisingly large share of datasets. However, this conclusion warrants closer inspection. Their method relied on the Wilcoxon signed-rank test using only 10-fold cross-validation samples. This approach has limited statistical power, and failing to find a "significant difference" is not the same as proving "no difference" [5]. Beyond that, the study offered scatter plots and meta-learning trends, but left practitioners without any concrete quantitative boundary rule they could actually use.

To move past the statistical power issue, this paper adopts a "practical significance framework." We define "linear sufficiency" as an acceptable accuracy drop (δ). Results are reported for δ=0.01,0.02,0.05, and the boundary's robustness is tested from 0 to 0.10. Using shallow decision trees of depth 3, we extract actionable conditional rules and validate them with leave-one-dataset-out cross-validation.

2. Related Work

2.1. Comparison of Linear and Non-linear Classifiers

Most studies comparing linear and non-linear classifiers originate from specific application domains—such as electricity forecasting, EEG signal classification, or credit scoring—and their conclusions seldom generalize beyond the field in which they were conducted. Few attempts have been made to establish a field-independent guideline. Post et al. (2016) [3] used OpenML to study how feature selection affects classification performance across 399 datasets, but their focus was on feature engineering rather than model complexity. Strang et al. (2018) [1] remains the largest systematic comparison in the "linear vs. non-linear" domain, covering 299 datasets with nested cross-validation and random search hyperparameter optimization.

2.2. Meta-learning and Algorithm Selection

Meta-learning approaches algorithm selection by transferring knowledge from historical tasks to guide choices on unseen datasets [7], [8]. The central difficulty is building a mapping from data characteristics (meta-features) to algorithm performance. Common features span simple, statistical, information-theoretic, and landmarker categories, but their predictive value varies considerably. Nambo et al. [6] applied genetic search to optimize these features, reaching 65.5% accuracy in predicting the best algorithm across 58 datasets. Maldonado et al. [17] took a different route, training convolutional neural networks directly on tabular data and outperforming traditional meta-feature methods in simulations. Strang et al. [1] showed that even simple meta-features can beat a majority-class baseline, though adding landmarker features comes at a notably higher computational cost.

2.3. Interpretability and Practical Significance

Linear models gain a distinct advantage under the GDPR's

"right to explanation" [4]. A recent case-law review by Kopf et al. [9] examined how this right—now embedded in both the GDPR and the AI Act—forces automated decision systems to be explainable. But interpretability is only one side of the problem. Choosing the right evaluation metric matters just as much. Traditional statistical tests, such as the Wilcoxon signed-rank test, often lose power when sample sizes are small [5]. A practical significance framework, which sets an acceptable loss threshold, aligns more directly with real-world decision-making needs than relying on p-values alone.

3. Methods

3.1. Data and Classifiers

Our experiments drew on the OpenML study_id=123 benchmark [1], [16]. After cleaning the data, we ended up with 1,687 usable evaluation records spanning 298 datasets. We tested three pairs of model families. For SVMs, we compared svm_linear (LinearSVC) against a kernel-based svm_rbf (SVC). For neural networks, a basic SGDClassifier (nn_linear) was pitted against a multi-layer perceptron (nn_mlp). The decision tree family included a single-split dt_stump (max_depth=1) and a full dt_full tree. Note that the dt_stump acts as our linear baseline here—it only splits once. On continuous data, this essentially turns it into a highly restricted linear classifier [11]. All evaluations relied on a 10-fold nested cross-validation. To tune the hyperparameters, we used 250 iterations of random search [12]–[14]. All models were built using scikit-learn.

Before training, the data underwent a specific preprocessing sequence: ConditionalImputer, OneHotEncoder, VarianceThreshold, and StandardScaler [10].

3.2. Practical Significance Framework

Since OpenML evaluation records only provide average accuracy, paired statistical testing is impossible. We adopt the practical significance framework [5]: defining "linear sufficiency" as the accuracy loss of the linear model not exceeding a threshold δ, i.e.:

$$diff = acc_{linear} - acc_{nonlinear} \geq -\delta \quad (1)$$

We report results for δ=0.01,0.02,0.05, and illustrate boundary robustness through continuous variations of δ from 0 to 0.10.

3.3. Boundary Sensitivity Analysis

For each classifier family, we calculated the curve of the "linear sufficiency" proportion as δ varies. This analysis does not rely on any meta-features; it relies solely on accuracy differences, making it universally applicable to all datasets.

3.4. Decision Rule Extraction and Validation

We used a decision tree of depth 3 to extract conditional rules for "linear sufficiency" (δ=0.02). The meta-features included: NumberOfInstances, NumberOfFeatures, FeatureToInstanceRatio, NumberOfClasses, PercentageOfMissingValues, DefaultAccuracy, and MinorityMajorityRatio. Rule performance was evaluated using leave-one-dataset-out cross-validation: for each dataset, the decision tree was trained on the remaining datasets and predicted on the held-out set, calculating accuracy, AUROC, and comparing against the majority-class baseline.

4. Results

4.1. Descriptive Statistics

Table 1 presents the basic statistics for the "linear sufficiency" proportions. Due to missing evaluation records for certain algorithms on specific datasets, the number of successfully paired datasets varies slightly across families (SVM: 289, NN: 218, DT: 271).

Table 1. Linear Sufficiency Proportions by Family (Practical Significance Framework)

Metric	SVM	NN	DT
Datasets	289	218	271
Absolute linear better (%)	48.4	40.8	32.5
δ=0.01 (%)	61.2	52.8	38.0
δ=0.02 (%)	68.9	62.4	46.1
δ=0.05 (%)	82.7	78.0	55.4

Non-linear models show superior performance on the majority of datasets across all three families. At δ=0.02, linear models are "sufficient" for 68.9% of SVM datasets, 62.4% of NN datasets, but only 46.1% of DT datasets. The proportion for the DT family is significantly lower than SVM and NN (χ^2 test, $p<0.001$), consistent with the representational limitations of decision stumps as minimalist linear models [11].

4.2. Boundary Sensitivity Analysis

As shown in Fig. 1, we observe how the sufficiency rate shifts as δ increases. All three families follow a similar trajectory: a steep initial climb that eventually flattens out. SVM and NN exceed the 50% threshold as soon as δ=0.02, whereas the DT family only crosses this boundary around δ=0.03. The DT curve remains noticeably lower and flatter, suggesting its performance is less responsive to relaxing the threshold. Even at δ=0.10, the DT sufficiency rate reaches just 66.4%, a stark contrast to the 92.0% and 89.0% achieved by SVM and NN, respectively.

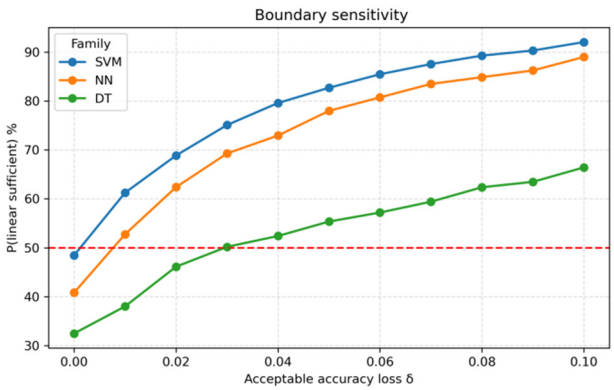


Fig 1. Boundary sensitivity curves

4.3. Decision Rule Extraction

We extracted decision trees of depth 3, as shown in Figs. 2-4. Univariate analysis reveals that the most discriminative features for the SVM family are NumberOfNumericFeatures (Mutual Information 0.094) and Dimensionality (0.089). For the NN family, NumberOfInstances and Dimensionality achieve mutual information of 0.105 and 0.087, respectively. The DT family relies most heavily on NumberOfFeatures (0.104) and NumberOfClasses (0.099).

Looking at the specific rules, the SVM model's root node splits on NumberOfFeatures <= 6752. When features are few, and further constrained by NumberOfClasses <= 2.5 and NumberOfInstances <= 53, the rule judges the linear model as "sufficient." The NN model starts with NumberOfClasses <= 2.5; for low-class-count data, it further splits on NumberOfInstances (at 106.5). The DT rule also starts with NumberOfClasses <= 2.5, subsequently judging "sufficient" at the NumberOfFeatures <= 24.5 node. The training set accuracies for these rules are 0.564 (SVM), 0.693 (NN), and 0.708 (DT), respectively.

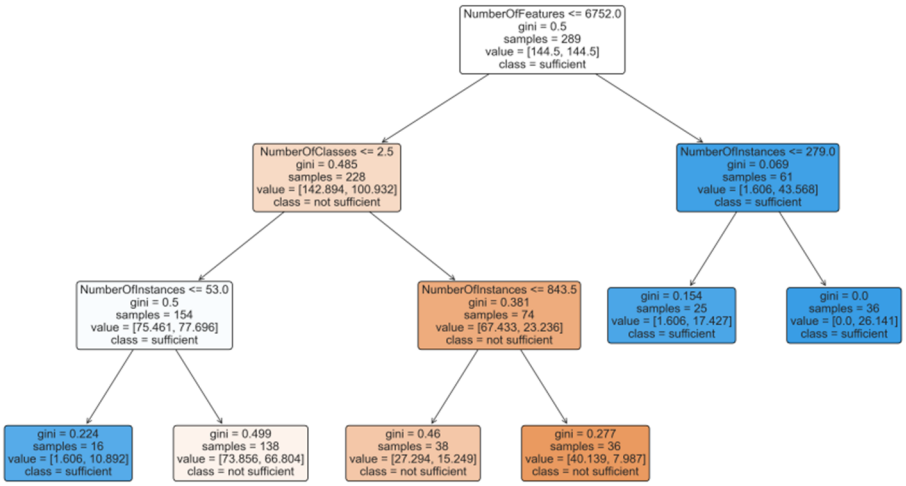


Fig 2. Decision tree with SVM rules

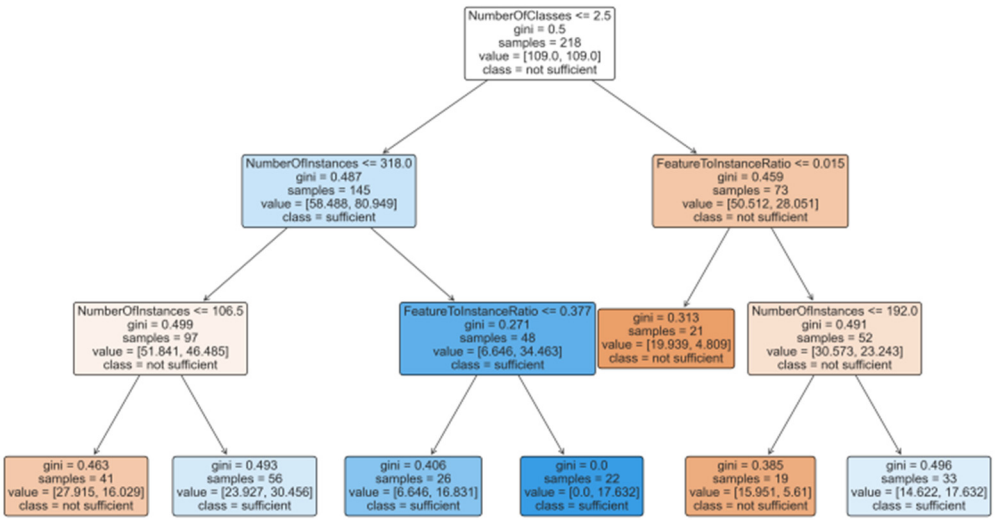


Fig 3. Decision tree with NN rules

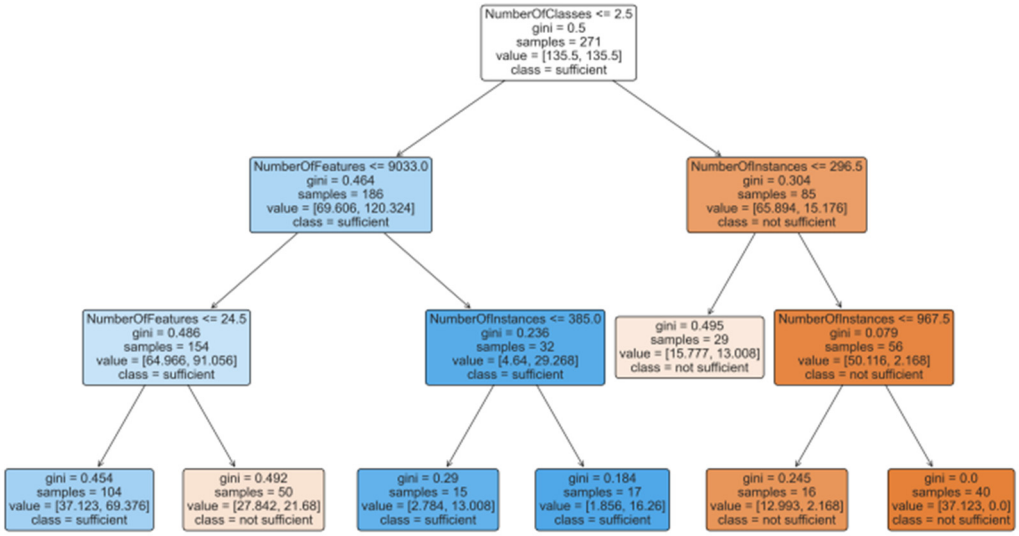


Fig 4. Decision tree with DT rules

4.4. Leave-One-Out Validation

Table II shows a notable contrast in the cross-validation results. The DT rules performed exceptionally well, achieving a 0.686 accuracy. This comfortably beats the 0.539 majority baseline by 14.8 percentage points and yields an AUROC of 0.642. Conversely, the SVM and NN rules performed poorly. Their accuracies dropped by 23.9 and 3.7 points below their respective baselines. With AUROCs near the 0.5 random level, these simple meta-features provide no reliable predictive advantage.

Table 2. Leave-One-Out Validation Performance

Metric	SVM	NN	DT
Samples	289	218	271
Accuracy	0.450	0.587	0.686
AUROC	0.515	0.485	0.642
Majority-class baseline	0.689	0.624	0.539
Improvement	−0.239	−0.037	**+0.148**

Therefore, while basic dataset features effectively predict linear sufficiency for decision trees, they are insufficient for constructing reliable selection rules for SVMs and neural networks.

5. Discussion

5.1. Main Findings

At δ=0.02, the gap between families is obvious: DT sits at just 46.1%, trailing far behind SVM (68.9%) and NN (62.4%). It is important not to conclude that non-linear trees are inherently better just because the DT score is low. A decision stump has a very narrow hypothesis space; it can only split along one axis. For continuous data, this essentially makes it a highly constrained linear model. Comparing a stump to a full tree fundamentally measures the jump from a minimalist model to a moderately complex one, rather than capturing the full spectrum of linear versus non-linear boundaries [11].

The boundary sensitivity curves highlight another key point. As we push δ from 0 to 0.10, all three families see their sufficiency rates rise. However, DT is the slowest to rise, requiring a much higher performance tolerance to catch up to the levels of SVM and NN.

A notable contrast is that the predictive value of simple meta-features for DT does not transfer well to SVM and NN. In leave-one-out validation, DT rules improved the baseline by 14.8 percentage points, whereas the SVM and NN rules were even worse than the naive "always predict sufficient" strategy. This likely indicates that the sufficiency conditions for SVM and NN are more complex than our current meta-features capture. They may depend more on local data structures, requiring richer meta-features (e.g., landmarker features) for effective characterization [16].

5.2. Limitations

Several limitations should be acknowledged. First, our practical significance approach depends on choosing a specific threshold δ. We ran sensitivity analyses to reduce this subjectivity, but picking the right δ ultimately still requires domain expertise. Another limitation comes from the data itself. Since the OpenML records only provide average accuracies, we could not run paired statistical tests. We had to infer from differences, which slightly reduces the statistical rigor [5]. Furthermore, the dt_stump we used is a highly restricted linear model. We cannot simply apply its boundary rules to other linear classifiers like Logistic Regression. Finally, our failure to beat the baseline with SVM and NN rules is a clear signal. It tells us that the current simple meta-features are insufficient to capture their complex boundaries. Future work will definitely require better feature engineering.

6. Conclusion

This study provided a quantitative characterization of the conditional boundaries for linear classifier sufficiency on large-scale OpenML benchmark data using a practical significance framework and interpretable rule extraction.

Our results confirm that non-linear models generally perform better across the board. However, the "linear sufficiency" rates vary dramatically depending on the model family. SVM exhibits the highest proportion (68.9% at δ=0.02), NN follows at 62.4%, and DT is the lowest at 46.1%. The leave-one-out validation reflects this disparity. While the shallow decision tree rules proved highly effective for DT—beating the 0.539 baseline with 0.686 accuracy—they completely failed to do the same for SVM and NN.

The data suggests that simple models should not be discarded outright. However, their viability heavily depends on the specific family and dataset features. Decision stumps have the narrowest applicability; linear SVM and linear neural networks are "sufficient" on roughly two-thirds of the datasets, making them worthwhile baselines in early modeling phases. Future work can proceed in two directions: first, introducing richer meta-features (e.g., landmarker features) to boost predictive performance for SVM and NN; second, exploring algorithm selection methods based on intrinsic data structures and extending the practical significance framework to other metrics like AUROC and F-measure.

References

[1] Strang, B., van der Putten, P., van Rijn, J. N., & Hutter, F. (2018). Don't rule out simple models prematurely: A large scale benchmark comparing linear and non-linear classifiers in OpenML. In Advances in intelligent data analysis XVII (pp. 303–315). Springer.

[2] Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. ACM SIGKDD Explorations Newsletter, 15(2), 49–60.

[3] Post, M. J., van der Putten, P., & van Rijn, J. N. (2016). Does feature selection improve classification? A large scale experiment in OpenML. In International Symposium on Intelligent Data Analysis (pp. 158–170). Springer.

[4] Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine, 38(3), 50–57.

[5] Dem?ar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7, 1–30.

[6] Nambo, H., Otsuka, A., Kimura, H., & Ueda, Y. (2016). Automatic classifier algorithm selection using optimized meta-features. Studies in Science and Technology, 5(2), 179–184.

[7] Rice, J. R. (1976). The algorithm selection problem. Advances in Computers, 15, 65–118.

[8] Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2009). Metalearning: Applications to data mining. Springer.

[9] Kopf, A., et al. (2024). The right to an explanation in practice: Insights from case law for the GDPR and the AI Act. In Proc. of the 2024 ACM Conference on Fairness, Accountability, and Transparency.

[10] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[11] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. Wadsworth.

[12] Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated machine learning: Methods, systems, challenges. Springer.

[13] van Rijn, J. N., & Hutter, F. (2017). Hyperparameter importance across datasets. In Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2367–2376). ACM.

[14] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in Neural Information Processing Systems (pp. 2962–2970).

[15] Sculley, D., Snoek, J., Wiltschko, A., & Rahimi, A. (2018). Winner's curse? On pace, progress, and empirical rigor. In Proc. of ICLR 2018.

[16] Bischl, B., Casalicchio, G., Feurer, M., Hutter, F., Lang, M., Mantovani, R. G., van Rijn, J. N., & Vanschoren, J. (2017). OpenML benchmarking suites and the OpenML100. arXiv preprint arXiv:1708.03731.

[17] Maldonado, S., Vairetti, C., & Figueroa, I. (2025). One-step learning algorithm selection for classification via convolutional neural networks. Information Sciences, 721, 122610.