WHEN DOES TRAINING BEAT PROMPTING ON TABULAR DATA?

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing between a prompted language model, a tabular foundation model, and a trained classical model requires understanding where their learning curves exchange advantage and how reproducible that boundary is. We study label-budget crossovers on 18 tabular datasets using 3 splits, shared evaluation rows, and training budgets from 8 to 2,048 labels. Equal-label comparisons are distinguished from comparisons against a fixed 32-shot LLM. We report observed crossover brackets, outcomes outside the measured range, and conditional stability under paired resampling of evaluation rows. For CatBoost classification against GPT-4.1-mini, median bootstrap retention is 57.8% for the exact bracket or censored endpoint, but 99.45% for the broad crossover category: uncertainty in location need not imply uncertainty in whether a crossing is observed. Model choice also changes qualitative outcomes. On the same split, CatBoost achieves lower error than 32-shot GPT-4.1-mini at some measured budget on all 8 regression datasets, but does not overtake 32-shot GPT-5.5 within the evaluated range on 5. A separate retrospective analysis of 126 implementations based on a shared starter protocol examines variation across implementations. These findings motivate reporting crossover location, outcome stability, and model dependence separately when assessing label efficiency in tabular prediction.

1 INTRODUCTION

At what label budget does a trained tabular model overtake a prompted language model? A crossing of two learning curves seems to answer this practical question, but it leaves two distinct uncertainties: whether a crossing occurs within the measured range, and where it occurs. A stable answer to the first can coexist with a poorly localised answer to the second. Moreover, both answers depend on the particular LLM being compared.

We separate these questions in a controlled comparison of prompted LLMs, TabPFN, and classical learners on 18 datasets. Methods share evaluation rows; the 8- and 32-label comparisons also share labelled training rows. Against a fixed 32-shot GPT-4.1-mini, CatBoost classification retains its observed crossover category in a median 99.45% of paired row-bootstrap draws, but its exact bracket or censored endpoint in 57.8%. Thus, a change in the reported label threshold need not reverse the qualitative finding. In contrast, changing the evaluated LLM can alter that finding: on one shared split, CatBoost is not observed to overtake GPT-5.5 on 5 of 8 regression datasets, compared with 0 for GPT-4.1-mini.

Our empirical contribution has three parts:

- **Location and outcome stability.** We distinguish observed crossover brackets, censored endpoints, and reversals, and quantify retention of both the broad outcome and the exact location under shared resampling.
- **Model dependence under explicit label accounting.** Equal-label comparisons establish the low-label reference; fixed-shot comparisons show how crossovers differ across classical and foundation models and two LLM arms evaluated on a common split.
- **Robustness and historical variation.** A common-evaluation duplicate sensitivity tests the observed boundaries. A separate cohort of 126 implementations with shared starter code

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

measures variation under a historical protocol, including a different classification scoring procedure.

Prior work already measures tabular label efficiency and few-shot prompting (Gardner et al., 2024; Grushina et al., 2026). Our target is the reproducibility of the comparison boundary, rather than a universal switching threshold or a new predictor. The historical cohort is complementary evidence under its own protocol; it does not isolate causal implementation effects or replicate the controlled benchmark's probability-based classification metric.

2 Related work

LLMs on serialised tabular data. Serialising rows into text and prompting a language model, with in-context exemplars standing in for training (Brown et al., 2020), has been studied for classification (Hegselmann et al., 2023), regression (Vacareanu et al., 2024; Requeima et al., 2024), language-interfaced fine-tuning (Dinh et al., 2022), feature engineering (Han et al., 2024), and boosting with prompted weak learners (Manikandan et al., 2023); Fang et al. (2024) survey the area. Tabular language models trained at scale, such as TP-BERTa (Yan et al., 2024) and TabuLa-8B (Gardner et al., 2024), adapt the backbone rather than freezing it. TabuLa-8B is the closest precursor to our setting: it compares a (fine-tuned) LLM against XGBoost and TabPFN at matched shot counts from one to thirty-two. Our focus is crossover location and outcome stability. More recently, Grushina et al. (2026) evaluate zero- and few-shot classification across real and synthetic suites, including repeated demonstration draws and prompt-information interventions. Our study separates retention of crossover categories from retention of their locations, includes regression, and examines a distinct historical cohort. Two strands of that literature shape our protocol. Few-shot prompting is sensitive to exemplar order and prompt wording (Lu et al., 2022; Zhao et al., 2021), which is why every serialisation choice here is frozen and why we treat implementation spread as a quantity to measure. And public tabular datasets can be memorised by LLMs (Bordt et al., 2024; Carlini et al., 2021; 2023), which is why the input-information ablations in Appendix H are named as sensitivity probes rather than contamination tests. Gorla & Puduppully (2026) show that evaluation construction and contamination can dominate apparent tabular transfer; our exact-record sensitivity addresses one limited aspect of evaluation construction. Controlled dimensionality experiments by Garnelo & Czarnecki (2026) examine explanations for LLM prediction failures, whereas our measurements do not identify an internal mechanism.

Tabular foundation models. Prior-fitted networks (Müller et al., 2022) condition on a labelled context set and predict without gradient updates; TabPFN (Hollmann et al., 2023; 2025) and its successors (Grinsztajn et al., 2025; 2026), TabICL (Qu et al., 2025), TabDPT (Ma et al., 2025), and CARTE (Kim et al., 2024) have improved rapidly. They consume labelled context without task-specific parameter updates. We evaluate two generations (Hollmann et al., 2025; Grinsztajn et al., 2026) under an identical protocol, on identical rows.

Gradient boosting versus deep models. Gradient-boosted trees (Chen & Guestrin, 2016; Prokhorenkova et al., 2018) have repeatedly been found to match or beat deep models on medium-sized tabular benchmarks (Gorishniy et al., 2021; Shwartz-Ziv & Armon, 2022; Grinsztajn et al., 2022; Borisov et al., 2024), with the gap narrowing under careful tuning (Kadra et al., 2021; Holzmüller et al., 2024) and depending on dataset properties (McElfresh et al., 2023). TabArena (Erickson et al., 2025) complicates the picture further: trees remain strong, deep models catch up with ensembling, and foundation models excel on small datasets. We therefore treat "trees dominate tabular learning" as an empirical question to be measured at each label budget rather than a premise, and use CatBoost with frozen defaults as the primary comparator. BeyondArena (Purucker et al., 2026) broadens evaluation to temporal and grouped tasks; our random-split results are confined to the evaluated IID-style setting.

Learning curves and crossover estimation. Learning-curve crossovers between model families were studied well before LLMs (Perlich et al., 2003), and learning curves are commonly summarised by power-law fits (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022) and extrapolated to plan sample sizes or terminate training (Mukherjee et al., 2003; Figueroa et al., 2012; Domhan et al., 2015). Reviews caution that power laws are one of several plausible shapes and that fitted

curves can be ill-behaved (Viering & Loog, 2023; Mohr & van Rijn, 2024). We report fits only as a secondary analysis (Appendix F), because the crossover implied by a fit is highly sensitive near the fitted floor and because extrapolation beyond the measured range is exactly where a crossover claim is weakest. Our primary statement is the measured bracket.

Replication and many-analyst studies. When independent teams analyse the same data, they reach materially different conclusions, in psychology (Silberzahn et al., 2018), neuroimaging (Botvinik-Nezer et al., 2020), the social sciences (Breznau et al., 2022; Schweinsberg et al., 2021), economics (Huntington-Klein et al., 2021; Menkveld et al., 2024), and ecology (Gould et al., 2025); multiverse and specification-curve analyses make that spread explicit (Steegen et al., 2016; Simonsohn et al., 2020). Machine learning has its own version of the problem: seeds, splits, and unreported choices move benchmark conclusions (Henderson et al., 2018; Bouthillier et al., 2021; Pineau et al., 2021), and leakage in preprocessing is a recurring source of irreproducible results (Kapoor & Narayanan, 2023). We study related variation in 126 implementations sharing a starter protocol. This cohort does not establish independence between analysts.

3 Label accounting and crossover outcomes

A prompted LLM consumes labelled exemplars; TabPFN conditions on a labelled context; a classical learner fits parameters to labelled rows. Both frozen model classes can consume labels without task-specific parameter updates. An *equal-label comparison* gives the methods the same labelled rows. A *fixed-shot comparison* instead holds the LLM's exemplars constant while increasing the comparator's labels. Its LLM error is horizontal because the recorded predictions are reused, not because LLM performance cannot improve with more exemplars.

Let e_L be the fixed-shot LLM error and $e_m(N_j)$ the comparator error at ordered measured budgets $N_1 < \cdots < N_J$. We record the first j such that $e_m(N_j) < e_L$. If $j > 1$, $(N_{j-1}, N_j]$ is the *observed bracket*; if $j = 1$, the comparator is *already better at the smallest budget*; if no such j exists, it is *not overtaken in range*. These are the three broad outcome categories. We separately record whether the advantage persists at all subsequent measured budgets and any later ordering changes.

A bracket describes an observed change on the grid, not a confidence interval or a unique population crossing. An endpoint category likewise describes the measured range. Resampling can preserve the broad category while moving its bracket. We quantify these two forms of stability separately, conditional on the fitted predictions, exemplars, and split.

4 Protocol

Datasets and splits. We use 18 curated tabular datasets (10 classification, 8 regression). Each is split $80/20$ under 3 seeds, stratified for classification. Every method is scored on *identical* held-out rows: within each split we freeze a capped evaluation sample of at most 500 rows before any model is run, so no method sees a different test set and no selection depends on measured performance.

Label budgets. Nested labelled subsets of 8, 32, 128, 512, 2,048 rows are drawn as prefixes of a *label-blind* permutation of the training split. Labels are never inspected to balance an exemplar set or to repair a degenerate subset; the permutation is a function of the split seed alone. Where the training pool is smaller than a nominal budget we use the entire pool, which happens for 8 of 18 datasets at the largest budget and is a source of censoring we report rather than hide. Plots and crossover brackets use realised label counts; aggregate budget tables use nominal caps. At 8 and 32 labels, the corresponding LLM's exemplars *are* the comparator's labelled subset, making those comparisons equal-label by construction.

Preprocessing. Imputation, encoding, and scaling are fitted on the budgeted training subset only and then applied to the evaluation rows. This matters: the historical version of this pipeline imputes over the full table before splitting, which leaks held-out information into training at every budget. Unseen categories at evaluation time encode as all-zero rather than triggering a refit.

Methods. The primary prompted arm, gpt-4.1-mini, is evaluated at 0, 8, and 32 shots on every split with one frozen key–value serialisation and a frozen exemplar ordering (Appendix B). Two further models, gpt-5.4 and gpt-5.5, are evaluated at 0 and 32 shots on one split as sensitivity checks. For models exposing token logprobs we take $P(y{=}1)$ from the normalised mass on the "0" and "1" tokens; where only one of the two appears in the returned top-k, we bound the absent token by the smallest probability actually returned, which avoids artificial $0/1$ saturation but is an approximation that need not preserve the ordering of the unobserved full-vocabulary probabilities. For a reasoning model that does not expose logprobs (gpt-5.5) we elicit a verbalised probability and report it as a distinct arm rather than pooling the two elicitations. One arm's logprob output proved degenerate after the fact (Appendix C); its classification results are excluded and its regression results, which do not depend on logprobs, are retained. Comparators are CatBoost (primary), TabPFN (two generations, v2 and TabPFN-3, without task-specific fine-tuning), random forest, and a linear model, all at frozen documented defaults, plus constant predictors as sanity checks.

Metrics. Classification uses AUROC computed from probabilities, with error defined as $1 - \mathrm{AUROC}$; balanced accuracy at a fixed 0.5 threshold is secondary. Regression uses RMSE, normalised by the standard deviation of the *fixed evaluation targets*. That denominator is shared across every method and budget within a split and is never supplied to a predictor, a selector, or a preprocessing step; it is confined to scoring. A zero-variance evaluation target makes the normalised metric undefined, and a single-class evaluation set makes AUROC undefined; both are recorded as undefined rather than imputed.

Degenerate subsets and failures. A labelled subset containing one class is not repaired. Supervised learners and the foundation models return the empirical constant class probability and the condition is recorded on every affected row (8 dataset–split–budget cells, 48 units, 2.67% of rows); the 8 corresponding 8-shot prompts, whose exemplars all carry one label, are sent to the LLM unchanged. A failed prediction is retried once with an identical request; rate limits and transient transport errors are retried separately with backoff and are not scored as model failures, while systematic access failures stop a run. Remaining failures fall back to the empirical class rate or mean target of that method's *own* allowed labelled subset, and to a fixed label-free constant for zero-shot calls. Across 672,627 stored predictions, 5 were model failures (0.0007%).

Uncertainty. Within a split we use 2,000 paired row bootstrap resamples that share row draws across methods and budgets, stratified for classification; these intervals are conditional on the fitted models and that split, and are computed for every unit (Appendix D). We also recompute crossovers within each paired resample. Aggregate means weight datasets equally and resample whole dataset blocks; cell counts and headroom distributions are descriptive. Generation comparisons first average paired differences over budgets and splits within each dataset, then use a dataset-block interval and a two-sided Wilcoxon signed-rank test over datasets. Neither budgets nor overlapping splits are treated as independent datasets.

5 Crossover outcomes, locations, and model choice

5.1 Equal-label reference

At 8 and 32 labels the LLM exemplars are exactly the comparator training rows. GPT-4.1-mini beats CatBoost in 26 and 24 of 30 classification cells, respectively; against TabPFN v2, the counts are 25 and 23. For regression, the corresponding counts are 20 and 10 of 24 against CatBoost, and 16 and 4 against TabPFN v2.

Table 1 adds effect sizes and paired dataset-block intervals at 32 labels, averaging splits within each dataset first. The mean classification advantage over CatBoost is 4.62 percentage points of AUROC. Regression intervals include zero for each comparator, despite different descriptive win counts. These contrasts concern the evaluated defaults and curated collection, and are not tuned performance envelopes. We do not treat splits as independent datasets or infer equivalence from intervals containing zero.

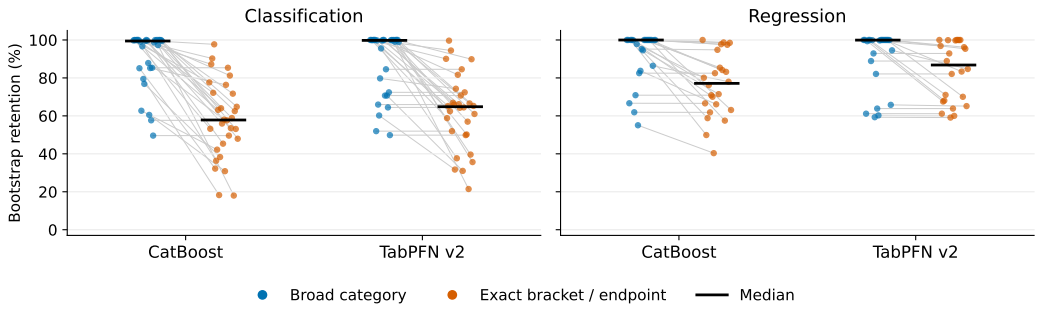


Figure 1: Conditional stability against the fixed 32-shot GPT-4.1-mini. Each connected pair is one dataset–split comparison: blue gives retention of the broad category, orange retention of the exact bracket or censored endpoint, and black marks the medians. Both use the same paired evaluation-row draws. A stable category can coexist with uncertain location. These frequencies condition on fitted predictions; they do not include retraining variation.

Table 1: Equal-label comparison at 32 labels. Differences are GPT-4.1-mini minus comparator error; negative favours the LLM. Split differences are averaged within dataset before the mean and a 20,000-draw dataset bootstrap. Intervals are exploratory; win counts are descriptive dataset–split counts.

Task	Comparator	Mean difference	95% interval	LLM wins
Classification	CatBoost	-0.0462	[-0.0814, -0.0158]	24/30
Classification	TabPFN v2	-0.0415	[-0.0816, -0.0096]	23/30
Classification	TabPFN v3	-0.0433	[-0.0851, -0.0092]	22/30
Regression	CatBoost	+0.0098	[-0.0730, +0.0884]	10/24
Regression	TabPFN v2	+0.0669	[-0.0066, +0.1389]	4/24
Regression	TabPFN v3	+0.0595	[-0.0243, +0.1384]	5/24

5.2 A STABLE CATEGORY DOES NOT LOCATE THE CROSSING

The remaining analyses hold GPT-4.1-mini at 32 shots unless stated otherwise; only the comparator's 32-label point is label-matched. Against CatBoost, 27 of 30 classification cells have a bracketed crossing and 3 are not overtaken in range. The regression counts are 21 bracketed and 3 already better at the smallest budget, with none not overtaken in range. No primary CatBoost comparison reverses. TabPFN v2 is already better at the smallest budget in 10 of 54 cells, but reverses in 4 classification cells. The full curves and categories are in Appendix D.

Figure 1 separates uncertainty in category from uncertainty in location. For CatBoost classification, median retention of the broad category is 99.45%, compared with 57.8% for the exact bracket or censored endpoint. TabPFN v2 likewise retains its category in 99.78% of draws at the median, versus 64.8% for location. Regression locations have higher median retention: 77.1% for CatBoost and 86.8% for TabPFN v2.

These are separate medians across cells, not necessarily statistics of the same cell. The distribution includes less stable comparisons even when its median category is stable. Near a small error margin, a bracket change need not imply a consequential prediction loss. The bootstrap measures conditional repeatability of the reported comparison; it does not quantify deployment regret or establish a unique population switching threshold. Censoring also matters: 8 datasets exhaust their training pool before the largest nominal cap, so endpoint categories concern the realised range.

5.3 THE EVALUATED LLM CAN CHANGE THE BROAD OUTCOME

The stronger-model sensitivity is qualitatively different from moving an adjacent bracket. On the common seed-42 split, CatBoost attains lower regression error than 32-shot GPT-4.1-mini at some measured budget on all 8 datasets. Against GPT-5.5 it is not observed to overtake on 5: Abalone, Ames Housing, Auto MPG, Diamonds, and Insurance Charges. TabPFN v2 and v3 are each not

Table 2: Datasets on which the comparator never attains lower error than the fixed 32-shot LLM within the evaluated range. Both LLM arms use seed 42 and identical evaluation rows; denominators are datasets, not independent splits. These are observed outcomes, not equivalence tests.

Task	Comparator	GPT-4.1-mini	GPT-5.5
Classification	CatBoost	1/10	3/10
Classification	TabPFN v2	1/10	2/10
Classification	TabPFN v3	0/10	3/10
Regression	CatBoost	0/8	5/8
Regression	TabPFN v2	0/8	4/8
Regression	TabPFN v3	0/8	4/8

observed to overtake on 4 datasets (Table 2). On classification, the GPT-5.5 comparison also includes a CatBoost reversal on German Credit, unlike the primary arm.

Both LLMs are evaluated on identical rows with the same 32 exemplars. These are comparisons between recorded API arms on one split, not a general effect of reasoning or a model-generation estimate. Classification additionally changes probability elicitation. The regression contrasts do not use that elicitation, but still require confirmation across splits. The TabPFN v2/v3 contrast and a hypothetical uniform-improvement sweep are secondary analyses in Appendices K and L; neither establishes a reliable generational improvement or forecasts a future LLM.

5.4 Sensitivity to repeated feature records

Distinct row identifiers do not guarantee distinct observations. In Wine Quality, 57–74 of the 500 evaluation rows per split match both features and target of a row in the largest used training subset. Repeated feature vectors also occur elsewhere (Appendix J). Their existence does not, by itself, prove real-world entity leakage.

We rescore every method and budget on a common subset excluding evaluation rows whose features match the largest used training subset. This exploratory sensitivity retains fitted predictions and changes the evaluation population; it is not a group-disjoint retraining experiment. None of the primary CatBoost, TabPFN v2, or v3 broad categories changes, although CatBoost's Wine Quality bracket moves from $(32, 128]$ to $(128, 512]$ on 2 splits. Against GPT-5.5 on Wine Quality, TabPFN v2 changes from bracketed to not overtaken in range. The 5 CatBoost regression non-crossings remain. This provides a concrete example of both location sensitivity and a model-dependent category change.

Cost context. At 32 shots, estimated uncached API cost is \$1.70 per thousand GPT-4.1-mini predictions and \$29.47 for GPT-5.5; median request latencies are 0.63 and 4.77 s. These token-based estimates exclude local compute, labelling, cache discounts, and unavailable discarded attempts (Appendix E).

6 Variation under a shared historical protocol

We analyze 126 usable implementations collected under a shared starter protocol over the same 18 datasets. The starter supplied data loading, serialization, prompting, querying, and aggregation helpers; participants completed specified experimental and analysis code. These implementations share substantial code and protocol choices.

The historical protocol differs materially from the controlled benchmark:

For binary hard predictions, AUROC equals balanced accuracy; it measures a different aspect of performance from probability ranking. Consequently, the historical cohort does not directly replicate the controlled classification estimand. It describes the spread in reported errors and fitted categories under its own procedure. Variation may combine API responses, evaluation sampling, and deviations in execution or analysis; these sources are not identified separately.

Table 3: The two studies have distinct evaluation procedures. Historical entries describe the supplied starter; individual submissions may deviate.

Aspect	Controlled benchmark	Historical starter
Classification score	AUROC from probabilities	AUROC from hard $0/1$ labels
Failed parsing	Declared fallback is scored	Unparsed rows omitted
LLM exemplars	0, 8, 32	0, 5
Evaluation sample	At most 500 rows	At most 100 rows
Comparator boundary	Observed grid	Pooled fitted reference curve

The median coefficient of variation of historical prediction error is 0.1476 (implementer-block bootstrap interval 0.0808–0.1704). Each implementer covers multiple assigned datasets; resampling identities preserves those records together, conditional on the fixed dataset collection and reference curves. It does not establish independence between implementers or propagate fitted-curve uncertainty.

Recomputing each implementation's crossover against the same pooled reference curve, 7.0% of implementation–comparison records (interval 4.7–9.5%) differ from the category implied by their cell's median error. The median majority share is 96.6%, while 60.7% of cells have at least one disagreement. The latter depends on cohort size and is descriptive. These figures quantify variation among shared-protocol submissions; they neither establish an irreducible noise floor nor isolate a causal effect of implementation choices (Appendix G).

7 Discussion and limitations

What the boundary tells us. Within the evaluated protocol, a crossover's broad outcome can be stable while its label-budget location is uncertain. The common-split GPT-5.5 comparison shows that the evaluated model can also change the broad outcome. Reporting categories, locations, and persistence separately makes these differences visible. A single label threshold would conceal them.

Coverage and dependence. The datasets are curated public tables with random splits, not a random sample of deployment tasks. Our findings do not establish performance under temporal or grouped distribution shift. Only GPT-4.1-mini is evaluated on all 3 splits; proprietary API aliases, a frozen prompt and exemplar policy, and comparator defaults limit generality. Public-data exposure may affect both language and tabular foundation models. The historical input-information probes do not isolate memorisation (Appendix H). The duplicate sensitivity changes the evaluated population without retraining and cannot replace grouped-split confirmation.

Execution and reproducibility. A retained TabPFN execution produced a substantial regression discrepancy without an exception; its cause and accepted-run repeatability remain unverified (Appendix M). The original probability parser and metadata recording also limit auditing of raw responses. Subsequent code fixes preserve failure flags, validate numeric responses, and isolate new runs, but do not alter or validate past outputs. Original dependency locks, checkpoint hashes, returned API snapshots, and system fingerprints were not retained. Saved predictions support recomputation of the reported analyses, not an exact fresh-execution guarantee.

Scope of practical conclusions. Conditional bootstrap frequencies are not simultaneous confidence guarantees across budgets and do not include retraining variability. A near-tie can move a bracket with little prediction cost. We do not evaluate a model-selection policy, and choosing the best method on these test outcomes would not establish deployment performance. Label cost and prediction volume would also be needed for an economic switching rule. The supported result is a characterization of comparison boundaries and their sensitivity, with the historical cohort interpreted under its separate scoring procedure.

AI USE STATEMENT

Generative AI assistants helped develop and critique the experimental protocol, implement the harness and data-processing code, analyse and interpret results, generate figures and tables, identify literature, and draft and edit the manuscript. AI-assisted review also identified and helped correct analysis and reporting errors. Numerical claims are generated from stored artifacts; protocol and analysis checks are executable in the accompanying code. The authors are responsible for the final design, interpretation, citations, and all AI-assisted content.

REPRODUCIBILITY STATEMENT

The protocol is specified in Section 4 and Appendices A–B; execution settings and reproduction limits are listed in Appendix I. The `bench/` harness generates tables, figures, and numerical macros from stored predictions and historical analysis artifacts. Row identifiers, data hashes, and conditional bootstrap summaries support recomputation without new API calls. The historical analysis and the controlled benchmark use distinct inputs and should be reproduced separately. The supplement provides historical aggregates without individual submissions or identifying records.

REFERENCES

Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. In *Conference on Language Modeling (COLM)*, 2024.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024. doi: 10.1109/TNNLS.2022.3229161.

Rotem Botvinik-Nezer, Felix Holzmeister, Colin F. Camerer, Anna Dreber, Juergen Huber, Magnus Johannesson, Michael Kirchler, et al. Variability in the analysis of a single neuroimaging dataset by many teams. *Nature*, 582(7810):84–88, 2020. doi: 10.1038/s41586-020-2314-9.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 3, 2021. arXiv:2103.03098.

Nate Breznau, Eike Mark Rinke, Alexander Wuttke, Hung H. V. Nguyen, Muna Adem, Jule Adriaans, et al. Observing many researchers using the same data and hypothesis reveals a hidden universe of uncertainty. *Proceedings of the National Academy of Sciences*, 119(44):e2203150119, 2022. doi: 10.1073/pnas.2203150119.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. arXiv:2005.14165.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 785–794, 2016.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009. doi: 10.1016/j.dss.2009.05.016.

Dean De Cock. Ames, iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011. doi: 10.1080/10691898.2011.11889627.

Robert Detrano, András Jánosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H. Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989. doi: 10.1016/0002-9149(89)90524-9.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. LIFT: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. arXiv:2506.16791.

Hadi Fanaee-T and João Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2–3):113–127, 2014. doi: 10.1007/s13748-013-0040-3.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models (LLMs) on tabular data: Prediction, generation, and understanding – a survey. *Transactions on Machine Learning Research*, 2024. arXiv:2402.17944.

Rosa L. Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H. Ngo. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12:8, 2012.

Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. arXiv:2406.12031.

Marta Garnelo and Wojciech M. Czarnecki. Why large language models fail at tabular prediction. arXiv preprint arXiv:2608.02412, 2026. URL `https://arxiv.org/abs/2608.02412`.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Aditya Gorla and Ratish Puduppully. The illusion of generalization in tabular language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2602.04031`.

Elliot Gould, Hannah S. Fraser, Timothy H. Parker, Shinichi Nakagawa, Simon C. Griffith, Peter A. Vesk, Fiona Fidler, et al. Same data, different analysts: Variation in effect sizes due to analytical decisions in ecology and evolutionary biology. *BMC Biology*, 23, 2025. doi: 10.1186/s12915-024-02101-x.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. TabPFN-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Daria Grushina, Kseniia Kuvshinova, Alina Kostromina, Aziz Temirkhanov, Mile Mitrovic, and Dmitry Simakov. LLMTabBench: Evaluating LLMs on binary tabular classification from zero to few shots. arXiv preprint arXiv:2605.24417, version 2, 2026. URL https://arxiv.org/abs/2605.24417.

Sungwon Han, Jinsung Yoon, Sercan O. Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2404.09491.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5549–5581, 2023.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. arXiv:1709.06560.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hans Hofmann. Statlog (german credit data). UCI Machine Learning Repository. https://doi.org/10.24432/C5NC77, 1994. Accessed 2026-07-27.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. arXiv:2407.04491.

Nick Huntington-Klein, Andreu Arenas, Emily Beam, Marco Bertoni, Jeffrey R. Bloem, Pralhad Burli, Naibin Chen, Paul Grieco, Godwin Ekpe, Todd Pugatch, Martin Saavedra, and Yaniv Stopnitzky. The influence of hidden researcher decisions in applied microeconomics. *Economic Inquiry*, 59(3):944–960, 2021. doi: 10.1111/ecin.12992.

IBM. IBM HR analytics employee attrition & performance. Kaggle; synthetic dataset created by IBM data scientists. https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset, 2017. Accessed 2026-07-27.

IBM. Telco customer churn. IBM Cognos Analytics sample data; distributed on Kaggle. https://www.kaggle.com/datasets/blastchar/telco-customer-churn, 2018. No canonical publication. Accessed 2026-07-27.

Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.16785.

Ron Kohavi. Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 202–207. AAAI Press, 1996.

Brett Lantz. *Machine Learning with R*. Packt Publishing, Birmingham, UK, 3rd edition, 2019. Source of the "insurance" (Medical Cost Personal) dataset; also distributed on Kaggle. `https://www.kaggle.com/datasets/mirichoi0218/insurance`. Accessed 2026-07-27.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. arXiv:2104.08786.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2410.18164.

Hariharan Manikandan, Yiding Jiang, and J. Zico Kolter. Language models are weak learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Albert J. Menkveld, Anna Dreber, Felix Holzmeister, Juergen Huber, Magnus Johannesson, Michael Kirchler, Sebastian Neusüß, Michael Razen, Utz Weitzel, et al. Nonstandard errors. *The Journal of Finance*, 79(3):2339–2390, 2024. doi: 10.1111/jofi.13337.

Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning: A survey. *Machine Learning*, 113:8371–8425, 2024. doi: 10.1007/s10994-024-06619-7.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014. doi: 10.1016/j.dss.2014.03.001.

Sayan Mukherjee, Pablo Tamayo, Simon Rogers, Ryan Rifkin, Anna Engle, Colin Campbell, Todd R. Golub, and Jill P. Mesirov. Estimating dataset size requirements for classifying DNA microarray data. *Journal of Computational Biology*, 10(2):119–142, 2003.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2112.10510.

Warwick J. Nash, Tracy L. Sellers, Simon R. Talbot, Andrew J. Cawthorn, and Wes B. Ford. The population biology of abalone (*Haliotis* species) in tasmania. i. blacklip abalone (*H. rubra*) from the north coast and islands of bass strait. Technical Report 48, Sea Fisheries Division, Marine Research Laboratories—Taroona, Department of Primary Industry and Fisheries, Tasmania, 1994. ISSN 1034-3288.

OpenAI. API model pricing. `https://developers.openai.com/api/docs/pricing`, 2026. Standard uncached rates checked September 17, 2026.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997. doi: 10.1016/S0167-7152(96)00140-X.

Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, 4:211–255, 2003.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research*, 22(164):1–20, 2021.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. arXiv:1706.09516.

Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond IID: How general are tabular foundation models, really? arXiv preprint arXiv:2606.30410, 2026. URL `https://arxiv.org/abs/2606.30410`.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.05564.

J. Ross Quinlan. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning (ICML)*, pp. 236–243. Morgan Kaufmann, 1993.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Duvenaud. LLM processes: Numerical predictive distributions conditioned on natural language. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. arXiv:2405.12856.

C. Okan Sakar, S. Olcay Polat, Mete Katircioglu, and Yomi Kastro. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. *Neural Computing and Applications*, 31(10):6893–6908, 2019. doi: 10.1007/s00521-018-3523-0.

Martin Schweinsberg, Michael Feldman, Nicola Staub, Olmo R. van den Akker, Robbie C. M. van Aert, Marcel A. L. M. van Assen, et al. Same data, different conclusions: Radical dispersion in empirical results when independent analysts operationalize and test the same hypothesis. *Organizational Behavior and Human Decision Processes*, 165:228–249, 2021. doi: 10.1016/j.obhdp.2021.02.003.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

Raphael Silberzahn, Eric L. Uhlmann, Daniel P. Martin, Pasquale Anselmi, Frederik Aust, Eli Awtrey, Štěpán Bahník, et al. Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1 (3):337–356, 2018. doi: 10.1177/2515245917747646.

Uri Simonsohn, Joseph P. Simmons, and Leif D. Nelson. Specification curve analysis. *Nature Human Behaviour*, 4(11):1208–1214, 2020. doi: 10.1038/s41562-020-0912-z.

Jack W. Smith, James E. Everhart, W. C. Dickson, William C. Knowler, and Robert S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 261–265. American Medical Informatics Association, 1988.

Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016. doi: 10.1177/1745691616658637.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.07544.

Tom Viering and Marco Loog. The shape of learning curves: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7799–7819, 2023.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2nd edition, 2016. ISBN 978-3-319-24277-4. `diamonds` dataset bundled with the ggplot2 R package. `https://ggplot2.tidyverse.org`.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z. Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2403.01841.

I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998. doi: 10.1016/S0008-8846(98)00165-3.

I-Cheng Yeh and Che-Hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009. doi: 10.1016/j.eswa.2007.12.020.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning (ICML)*, 2021. arXiv:2102.09690.

A Dataset and split details

Table 4 lists the 18 datasets with their sizes, the frozen splits, and the realised label budgets. Each dataset is split $80/20$ under seeds 42, 43, 44 (stratified for classification); the evaluation sample is the full held-out split when it has at most 500 rows and a frozen stratified (classification) or uniform (regression) sample of 500 rows otherwise. Nested labelled subsets are prefixes of a label-blind permutation of the training split, so the 8-row subset is contained in the 32-row subset and so on. Auto MPG and Heart Disease (2 datasets) have training pools below the fourth nominal budget and therefore fewer measured budgets; 8 datasets use the whole pool at the largest nominal budget. Content hashes of every raw file and of every frozen evaluation sample are recorded in the stored manifest.

Table 4: Datasets, frozen splits, and realised label budgets. "Pool" is the training split (seed 42; the two other seeds differ by at most one row); "Eval" is the frozen capped evaluation sample scored by every method. "Labels" lists the realised labelled-subset sizes at the nominal budgets 8, 32, 128, 512, 2,048 – a pool smaller than a nominal budget is used whole and larger nominal budgets collapse onto it. "1-class" counts units (seed × budget) whose label-blind subset contained a single class.

Dataset	Task	Rows	Feat.	Pool	Eval	Labels at budgets	1-class
Abalone (Nash et al., 1994)	reg	4,177	8	3,341	500	8/32/128/512/2048	0
Adult Income (Kohavi, 1996)	cls	45,222	14	36,177	500	8/32/128/512/2048	1
Ames Housing (De Cock, 2011)	reg	1,460	74	1,168	292	8/32/128/512/1168	0
Auto MPG (Quinlan, 1993)	reg	398	7	318	80	8/32/128/318	0
Bank Marketing (Moro et al., 2014)	cls	45,211	16	36,168	500	8/32/128/512/2048	3
Bike Sharing (Fanaee-T & Gama, 2014)	reg	17,379	12	13,903	500	8/32/128/512/2048	0
California Housing (Pace & Barry, 1997)	reg	20,640	8	16,512	500	8/32/128/512/2048	0
Concrete Strength (Yeh, 1998)	reg	1,030	8	824	206	8/32/128/512/824	0
Credit Card Default (Yeh & Lien, 2009)	cls	30,000	23	24,000	500	8/32/128/512/2048	0
Diabetes (Pima) (Smith et al., 1988)	cls	768	8	614	154	8/32/128/512/614	0
Diamonds (Wickham, 2016)	reg	53,940	9	43,152	500	8/32/128/512/2048	0
Employee Attrition (IBM, 2017)	cls	1,470	30	1,176	294	8/32/128/512/1176	2
German Credit (Hofmann, 1994)	cls	1,000	20	800	200	8/32/128/512/800	0
Heart Disease (Detrano et al., 1989)	cls	303	13	242	61	8/32/128/242	0
Insurance Charges (Lantz, 2019)	reg	1,338	6	1,070	268	8/32/128/512/1070	0
Online Shoppers (Sakar et al., 2019)	cls	12,330	17	9,864	500	8/32/128/512/2048	1
Telco Churn (IBM, 2018)	cls	7,043	19	5,634	500	8/32/128/512/2048	1
Wine Quality (Cortez et al., 2009)	cls	6,497	12	5,197	500	8/32/128/512/2048	0

B Prompt templates and probability extraction

Listing 1 reproduces the frozen system messages and the structure of the user message, generated directly from the harness code for one real evaluation row. Rows are serialised as `key: value` lines in the dataset's column order, with missing values written as `missing`; exemplars appear in the label-blind permutation order and are never reordered or balanced by label. The dataset and target descriptions and the per-feature documentation come from the dataset metadata shipped with the code.

```
# system message (classification, logprob arms)
You are a careful tabular classifier. Reply with exactly one character: 0
     or 1. No other text.

# system message (classification, verbalised arm)
You are a careful tabular classifier. Reply with ONLY a probability
    between 0 and 1 (e.g. 0.73) that the target equals 1. No other text.

# system message (regression)
You are a careful tabular regressor. Reply with ONLY a number, the
    predicted target value. No units, no other text.

# user message (heart_disease, seed 42, 1 exemplar shown; the 8- and 32-
    shot
# prompts list the first 8 / 32 rows of the same label-blind permutation)
Dataset: Predicts heart disease presence from clinical measurements.
    Cleveland dataset, UCI.
Target: 1 = heart disease present, 0 = absent
Features:
```

```
- <one line per documented feature>

Example 1:
age: 59.0
sex: 1.0
cp: 1.0
trestbps: 160.0
chol: 273.0
fbs: 0.0
restecg: 2.0
thalach: 125.0
exang: 0.0
oldpeak: 0.0
slope: 1.0
ca: 0.0
thal: 3.0
Answer: 1

Now predict:
age: 37.0
sex: 1.0
cp: 3.0
trestbps: 130.0
chol: 250.0
fbs: 0.0
restecg: 0.0
thalach: 187.0
exang: 0.0
oldpeak: 3.5
slope: 3.0
ca: 0.0
thal: 3.0
Answer:
```

Listing 1: Frozen system messages and one-exemplar user message generated from the harness for Heart Disease (seed 42).

Probability extraction. For arms exposing token logprobs we request the top-5 alternatives for the first generated token and take $P(y{=}1) = p_1/(p_0 + p_1)$, where p_0 and p_1 are the probabilities of the tokens `0` and `1`. When only one of the two tokens appears in the returned top-5, the absent token is assigned the smallest probability actually returned, which is an upper bound on any token outside the top-5. This avoids saturation caused by treating absent mass as zero, but the substitution can change probability rankings and is not a recovered exact class probability. The rule has a known degenerate case: if the API returns a single entry, the floor equals the present token's probability and $P(y{=}1)$ is exactly 0.5. Exactly 0.5 probabilities occur on 0.43% of gpt-4.1-mini classification rows (at most 2.0% in any unit); this tie count alone does not identify the mechanism on each row. For gpt-5.4, ties affect almost every row (Appendix C). For the arm without logprobs, a verbalised probability is parsed from the reply. The original regression runner parsed the first number in the text. A later audit found that its parser mishandled leading-decimal replies and accepted nonfinite numbers. Original raw responses were not retained for a complete re-audit. The revised runner validates complete finite numeric replies and rejects out-of-range probabilities; these changes apply only to new runs.

C GPT-5.4 CLASSIFICATION: A PROBABILITY-EXTRACTION DEFECT

The gpt-5.4 arm was run with the same logprob extraction as gpt-4.1-mini. On inspection of the stored predictions, 91.95% of its classification rows carry $P(y{=}1)$ of exactly 0.5 (up to 97.6% in a single unit; Table 13). A diagnostic request confirmed the mechanism: for this model the API returns only the sampled token in `top_logprobs` regardless of the requested k, so the bounding rule ties the two class tokens on every confident row. Because the stored hard label is derived from the probability ($\hat{y} = \mathbb{1}[P \geq 0.5]$), the hard prediction is lost as well and the 20 classification units

cannot be rescored from the prediction tables. Their AUROC is near 0.5 by construction, not because of the model, and we do not report them anywhere in this paper. The regression units of the same arm parse a number from the reply, are unaffected, and are reported in Tables 6 and 8. We record this here rather than silently dropping the arm because it is a third instance (after the preprocessing leak and the accelerator incident of Appendix M) of a failure that produced plausible-looking numbers with no error raised; the fix, had we caught it before running, would have been to elicit a verbalised probability for this model as we did for gpt-5.5.

D PER-METHOD RESULTS

Tables 5 and 6 give the aggregate learning curves with dataset-block intervals; Tables 9 and 10 give every (dataset, method, budget) cell averaged over splits; Tables 7 and 8 give the prompted-LLM arms per dataset, including the two models run on a single split as sensitivity checks; and Table 11 lists the crossover category of every primary (dataset, split, comparator) cell. Paired row-bootstrap intervals (3 splits $\times$ 18 datasets, 2,000 resamples sharing row draws across methods and budgets) are computed for every one of the 1818 units and stored as an artifact; they are quoted in Tables 7 and 8 for the single-split arms, where they are the only uncertainty available.

Two observations from these tables are worth stating because the main text relies on them. First, on classification, adding exemplars to gpt-4.1-mini barely moves its error: the median across (dataset, split) cells is 0.183 at zero shots, 0.204 at eight, and 0.199 at thirty-two, and the 32-shot prompt beats the zero-shot prompt in only 53.3% of 30 cells (median paired change -0.0014). On regression the exemplars matter more (87.5% of 24 cells; median change -0.0778), largely because zero-shot numeric guesses can be off by orders of magnitude on some targets. Second, the newer reasoning model is better than gpt-4.1-mini on the same rows: at 32 shots on the seed-42 split, gpt-5.5 has lower error on 80% of classification datasets (median paired change -0.042) and on 100% of regression datasets (median change -0.349); gpt-5.4 also has lower error on 100% of regression datasets (median change -0.224). These are single-split contrasts between recorded aliases of proprietary models and are reported as sensitivity checks, not as a population estimate over frontier systems. Section 5 uses gpt-4.1-mini for the three-split analysis and contrasts the two model arms on their common split.

Table 5: Classification: mean error across datasets by method and label budget (each dataset averaged over its three splits first), with 95% intervals from resampling whole dataset blocks. Columns are nominal caps; saturated pools are carried forward so every comparator column includes the same datasets. Fixed-shot LLM rows are listed at the budget equal to their exemplar count. Error is $1 - \mathrm{AUROC}$ for classification and normalised RMSE for regression.

Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
CatBoost	0.368 [0.309, 0.423]	0.252 [0.196, 0.304]	0.186 [0.151, 0.222]	0.155 [0.122, 0.188]	0.137 [0.107, 0.167]
TabPFN v2	0.350 [0.285, 0.405]	0.247 [0.190, 0.306]	0.181 [0.148, 0.215]	0.147 [0.117, 0.179]	0.130 [0.101, 0.160]
TabPFN-3	0.356 [0.291, 0.412]	0.249 [0.189, 0.315]	0.180 [0.147, 0.216]	0.148 [0.118, 0.179]	0.128 [0.098, 0.158]
Random forest	0.354 [0.288, 0.409]	0.253 [0.203, 0.301]	0.193 [0.158, 0.227]	0.167 [0.140, 0.196]	0.149 [0.121, 0.176]
Linear model	0.368 [0.312, 0.413]	0.274 [0.233, 0.322]	0.208 [0.180, 0.240]	0.171 [0.140, 0.210]	0.160 [0.130, 0.195]
Constant	0.500 [0.500, 0.500]	0.500 [0.500, 0.500]	0.500 [0.500, 0.500]	0.500 [0.500, 0.500]	0.500 [0.500, 0.500]
gpt-4.1-mini 0-shot	0.217 [0.161, 0.297]†				
gpt-4.1-mini 8-shot	0.210 [0.156, 0.278]				
gpt-4.1-mini 32-shot		0.206 [0.158, 0.262]			
gpt-5.5 0-shot	0.153 [0.112, 0.191]†				
gpt-5.5 32-shot		0.147 [0.104, 0.189]			

†zero-shot rows use no labels and are placed in the first column for layout only. gpt-5.4 and gpt-5.5 were run on the seed-42 split only, so their intervals reflect dataset resampling of single-split estimates; gpt-5.4 classification is omitted (Appendix C).

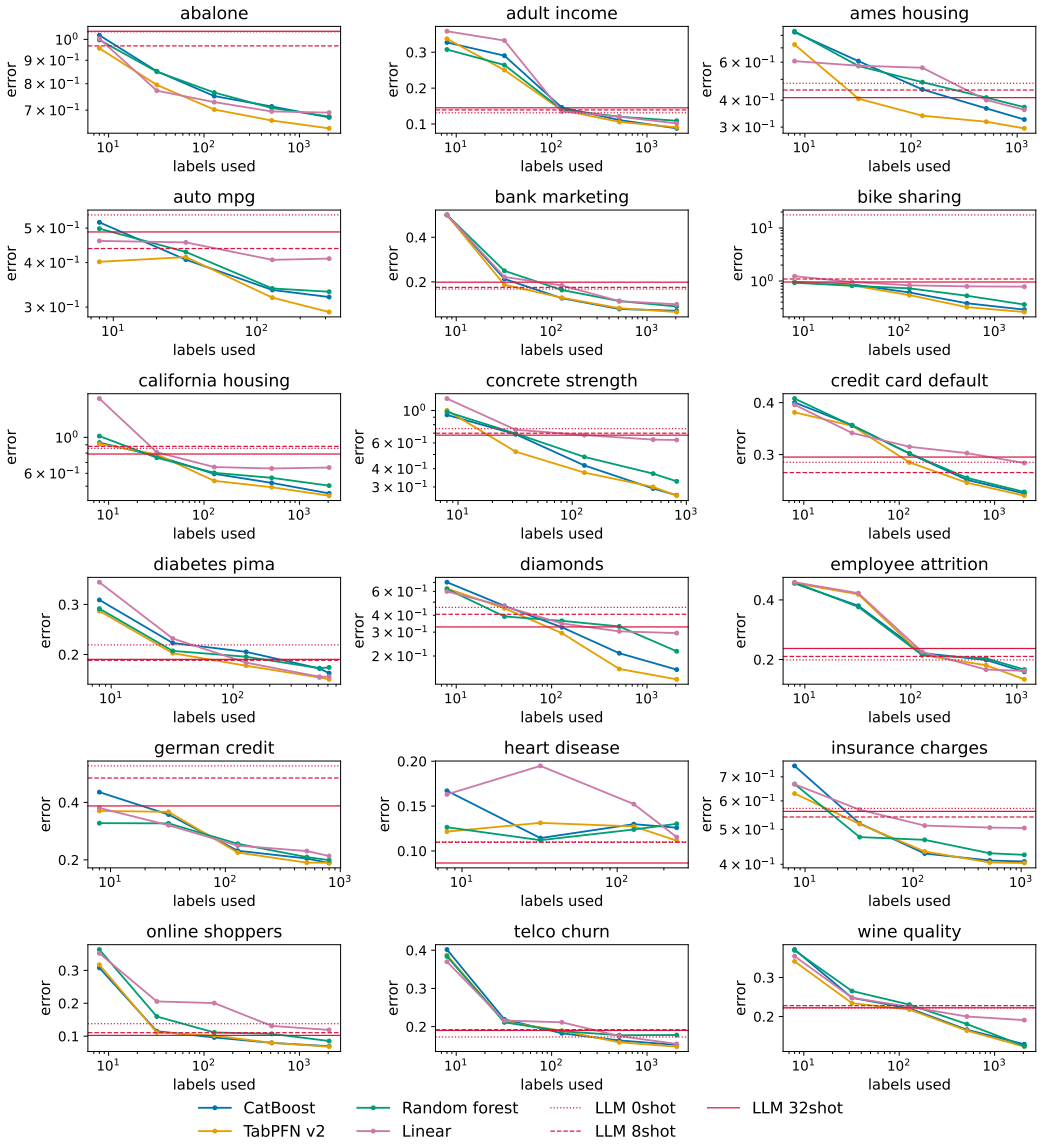


Figure 2: Error against label budget for every dataset. Comparator learning curves are solid; horizontal lines mark fixed-shot LLM arms, which are flat in budget because their exemplar count is fixed. Classification error is $1 - \mathrm{AUROC}$; regression error is RMSE normalised by the evaluation-target standard deviation (logarithmic vertical axes for regression). Curves average the three splits and use realised label counts. Axes are per-dataset: the quantity of interest is where curves cross a flat line, not the absolute heights.

Table 6: Regression: mean error across datasets by method and label budget (each dataset averaged over its three splits first), with 95% intervals from resampling whole dataset blocks. Columns are nominal caps; saturated pools are carried forward so every comparator column includes the same datasets. Fixed-shot LLM rows are listed at the budget equal to their exemplar count. Error is $1 - \text{AUROC}$ for classification and normalised RMSE for regression.

Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
CatBoost	0.830 [0.713, 0.932]	0.647 [0.532, 0.758]	0.489 [0.398, 0.591]	0.402 [0.313, 0.507]	0.361 [0.270, 0.469]
TabPFN v2	0.776 [0.632, 0.909]	0.590 [0.476, 0.717]	0.443 [0.360, 0.542]	0.369 [0.281, 0.475]	0.340 [0.250, 0.444]
TabPFN-3	0.787 [0.638, 0.923]	0.597 [0.488, 0.714]	0.428 [0.357, 0.517]	0.361 [0.278, 0.456]	0.333 [0.247, 0.434]
Random forest	0.819 [0.682, 0.941]	0.623 [0.508, 0.738]	0.529 [0.429, 0.637]	0.459 [0.382, 0.556]	0.402 [0.320, 0.498]
Linear model	0.939 [0.687, 1.239]	0.668 [0.552, 0.788]	0.591 [0.486, 0.696]	0.548 [0.437, 0.655]	0.541 [0.432, 0.649]
Constant	1.052 [1.033, 1.073]	1.017 [1.009, 1.028]	1.003 [1.002, 1.004]	1.002 [1.001, 1.003]	1.002 [1.001, 1.003]
gpt-4.1-mini 0-shot	2.795 [0.575, 7.078]†				
gpt-4.1-mini 8-shot	0.684 [0.513, 0.857]				
gpt-4.1-mini 32-shot		0.657 [0.500, 0.823]			
gpt-5.4 0-shot	0.479 [0.364, 0.592]†				
gpt-5.4 32-shot		0.443 [0.326, 0.564]			
gpt-5.5 0-shot	0.313 [0.217, 0.406]†				
gpt-5.5 32-shot		0.303 [0.201, 0.417]			

†zero-shot rows use no labels and are placed in the first column for layout only. gpt-5.4 and gpt-5.5 were run on the seed-42 split only, so their intervals reflect dataset resampling of single-split estimates; gpt-5.4 classification is omitted (Appendix C).

Table 7: Classification: prompted-LLM error per dataset. gpt-4.1-mini: mean ± SD over three splits. gpt-5.5: seed-42 split only, with the paired row-bootstrap 95% interval conditional on that split. gpt-5.4 classification is not reported (Appendix C).

Dataset	4.1-mini 0-shot	4.1-mini 8-shot	4.1-mini 32-shot	5.5 0-shot	5.5 32-shot
Adult Income	0.131±0.010	0.139±0.019	0.145±0.017	0.112 [0.083, 0.144]	0.114 [0.083, 0.146]
Bank Marketing	0.168±0.045	0.176±0.048	0.199±0.044	0.084 [0.051, 0.120]	0.082 [0.051, 0.116]
Credit Card Default	0.285±0.023	0.265±0.028	0.295±0.043	0.260 [0.203, 0.319]	0.271 [0.213, 0.333]
Diabetes (Pima)	0.219±0.044	0.188±0.052	0.190±0.042	0.172 [0.111, 0.240]	0.139 [0.086, 0.201]
Employee Attrition	0.199±0.013	0.211±0.021	0.237±0.010	0.194 [0.124, 0.273]	0.205 [0.132, 0.285]
German Credit	0.527±0.010	0.485±0.016	0.388±0.043	0.224 [0.157, 0.290]	0.203 [0.139, 0.268]
Heart Disease	0.110±0.056	0.110±0.047	0.087±0.053	0.033 [0.001, 0.078]	0.021 [0.001, 0.053]
Online Shoppers	0.138±0.013	0.111±0.018	0.102±0.020	0.093 [0.054, 0.140]	0.094 [0.054, 0.143]
Telco Churn	0.173±0.009	0.192±0.020	0.190±0.009	0.160 [0.124, 0.197]	0.160 [0.123, 0.198]
Wine Quality	0.222±0.013	0.228±0.005	0.222±0.017	0.194 [0.147, 0.244]	0.181 [0.133, 0.232]

Table 8: Regression: prompted-LLM error per dataset. gpt-4.1-mini: mean ± SD over three splits. gpt-5.5 and gpt-5.4: seed-42 split only, with the paired row-bootstrap 95% interval conditional on that split.

Dataset	4.1-mini 0-shot	4.1-mini 8-shot	4.1-mini 32-shot	5.5 0-shot	5.5 32-shot	5.4 0-shot	5.4 32-shot
Abalone	1.039±0.052	0.968±0.070	1.042±0.207	0.509 [0.422, 0.595]	0.592 [0.519, 0.666]	0.783 [0.716, 0.855]	0.767 [0.688, 0.849]
Ames Housing	0.479±0.055	0.446±0.038	0.411±0.033	0.238 [0.200, 0.273]	0.222 [0.189, 0.255]	0.326 [0.251, 0.410]	0.307 [0.242, 0.378]
Auto MPG	0.544±0.064	0.438±0.047	0.488±0.065	0.107 [0.022, 0.171]	0.130 [0.015, 0.203]	0.365 [0.301, 0.426]	0.306 [0.243, 0.369]
Bike Sharing	17.666±0.986	1.091±0.339	0.957±0.097	0.349 [0.297, 0.409]	0.303 [0.260, 0.347]	0.493 [0.436, 0.549]	0.553 [0.484, 0.623]
California Housing	0.855±0.029	0.879±0.111	0.789±0.043	0.469 [0.400, 0.541]	0.492 [0.420, 0.566]	0.663 [0.589, 0.733]	0.599 [0.521, 0.674]
Concrete Strength	0.751±0.032	0.699±0.044	0.678±0.052	0.412 [0.345, 0.486]	0.364 [0.319, 0.411]	0.518 [0.471, 0.565]	0.401 [0.358, 0.444]
Diamonds	0.457±0.010	0.407±0.099	0.328±0.019	0.156 [0.132, 0.184]	0.156 [0.130, 0.182]	0.242 [0.208, 0.278]	0.238 [0.191, 0.280]
Insurance Charges	0.572±0.069	0.541±0.062	0.561±0.028	0.264 [0.171, 0.346]	0.164 [0.004, 0.253]	0.441 [0.352, 0.527]	0.374 [0.284, 0.451]

Table 9: Classification: per-dataset error by method and label budget, mean ± SD across the three splits (SD over three values; a descriptive spread, not a confidence interval). Where a dataset's pool is smaller than a nominal budget the whole pool was used (Table 4). Paired row-bootstrap intervals for every (dataset, split, method, budget) unit are in the analysis artifact `row_bootstrap.parquet`.

Dataset	Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
Adult Income	CatBoost	0.327±0.150	0.290±0.084	0.146±0.002	0.111±0.011	0.088±0.012
	TabPFN v2	0.337±0.141	0.250±0.083	0.139±0.006	0.106±0.005	0.091±0.012
	TabPFN-3	0.342±0.137	0.263±0.092	0.137±0.008	0.112±0.004	0.095±0.010
	Random forest	0.308±0.169	0.264±0.133	0.140±0.003	0.120±0.016	0.109±0.006
	Linear model	0.358±0.123	0.333±0.222	0.135±0.015	0.120±0.003	0.102±0.012
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000

Dataset	Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
Bank Marketing	CatBoost	0.500±0.000	0.214±0.035	0.127±0.027	0.079±0.007	0.071±0.009
	TabPFN v2	0.500±0.000	0.187±0.012	0.130±0.023	0.083±0.017	0.065±0.015
	TabPFN-3	0.500±0.000	0.189±0.011	0.127±0.018	0.088±0.007	0.066±0.012
	Random forest	0.500±0.000	0.250±0.050	0.164±0.048	0.114±0.009	0.090±0.007
	Linear model	0.500±0.000	0.222±0.026	0.184±0.049	0.114±0.020	0.100±0.013
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Credit Card Default	CatBoost	0.400±0.152	0.357±0.067	0.302±0.042	0.251±0.019	0.225±0.021
	TabPFN v2	0.381±0.159	0.356±0.085	0.285±0.051	0.246±0.027	0.221±0.038
	TabPFN-3	0.408±0.165	0.366±0.081	0.291±0.058	0.246±0.024	0.214±0.026
	Random forest	0.408±0.175	0.356±0.075	0.302±0.028	0.255±0.008	0.228±0.012
	Linear model	0.396±0.102	0.342±0.039	0.315±0.029	0.303±0.019	0.284±0.025
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Diabetes (Pima)	CatBoost	0.309±0.097	0.223±0.069	0.205±0.049	0.172±0.020	0.163±0.031
	TabPFN v2	0.287±0.140	0.203±0.079	0.178±0.041	0.154±0.019	0.150±0.030
	TabPFN-3	0.322±0.092	0.211±0.075	0.185±0.037	0.156±0.019	0.154±0.031
	Random forest	0.292±0.099	0.207±0.054	0.196±0.036	0.173±0.020	0.174±0.021
	Linear model	0.344±0.173	0.232±0.081	0.184±0.046	0.155±0.026	0.156±0.034
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Employee Attrition	CatBoost	0.455±0.077	0.380±0.135	0.221±0.038	0.199±0.040	0.159±0.027
	TabPFN v2	0.456±0.076	0.419±0.177	0.215±0.045	0.181±0.034	0.134±0.026
	TabPFN-3	0.458±0.073	0.423±0.184	0.225±0.048	0.188±0.035	0.139±0.020
	Random forest	0.458±0.072	0.377±0.124	0.215±0.025	0.204±0.028	0.166±0.023
	Linear model	0.459±0.070	0.423±0.167	0.225±0.048	0.166±0.022	0.162±0.023
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
German Credit	CatBoost	0.436±0.030	0.358±0.060	0.232±0.027	0.205±0.037	0.189±0.040
	TabPFN v2	0.370±0.014	0.367±0.021	0.225±0.032	0.190±0.029	0.190±0.036
	TabPFN-3	0.389±0.058	0.357±0.015	0.230±0.032	0.196±0.035	0.191±0.039
	Random forest	0.328±0.060	0.327±0.030	0.256±0.028	0.210±0.029	0.199±0.033
	Linear model	0.381±0.039	0.321±0.040	0.249±0.011	0.231±0.047	0.213±0.046
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Heart Disease	CatBoost	0.167±0.063	0.114±0.033	0.130±0.071	0.126±0.075	–
	TabPFN v2	0.122±0.026	0.131±0.026	0.127±0.067	0.111±0.066	–
	TabPFN-3	0.135±0.032	0.117±0.047	0.118±0.065	0.101±0.065	–
	Random forest	0.126±0.030	0.112±0.029	0.124±0.072	0.130±0.069	–
	Linear model	0.163±0.066	0.195±0.042	0.152±0.068	0.115±0.064	–
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	–
Online Shoppers	CatBoost	0.308±0.167	0.115±0.017	0.096±0.025	0.080±0.012	0.070±0.009
	TabPFN v2	0.317±0.163	0.111±0.020	0.102±0.015	0.080±0.007	0.068±0.009
	TabPFN-3	0.268±0.202	0.113±0.017	0.097±0.009	0.079±0.011	0.065±0.010
	Random forest	0.363±0.139	0.160±0.024	0.111±0.009	0.107±0.007	0.085±0.013
	Linear model	0.352±0.134	0.206±0.010	0.201±0.054	0.131±0.015	0.119±0.011
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Telco Churn	CatBoost	0.402±0.098	0.219±0.010	0.183±0.013	0.164±0.016	0.152±0.015
	TabPFN v2	0.387±0.107	0.213±0.018	0.189±0.019	0.159±0.010	0.148±0.010
	TabPFN-3	0.408±0.082	0.208±0.018	0.186±0.025	0.157±0.011	0.147±0.010
	Random forest	0.384±0.121	0.211±0.033	0.188±0.007	0.178±0.020	0.178±0.015
	Linear model	0.370±0.169	0.216±0.008	0.211±0.020	0.175±0.008	0.155±0.008
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
Wine Quality	CatBoost	0.371±0.097	0.248±0.022	0.221±0.031	0.166±0.009	0.129±0.002
	TabPFN v2	0.341±0.068	0.234±0.024	0.218±0.034	0.164±0.010	0.123±0.003
	TabPFN-3	0.326±0.066	0.242±0.020	0.208±0.013	0.154±0.008	0.108±0.005
	Random forest	0.369±0.114	0.265±0.035	0.230±0.031	0.181±0.012	0.124±0.013
	Linear model	0.354±0.086	0.248±0.045	0.225±0.021	0.200±0.016	0.191±0.009
	Constant	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000

Table 10: Regression: per-dataset error by method and label budget, mean ± SD across the three splits (SD over three values; a descriptive spread, not a confidence interval). Where a dataset's pool is smaller than a nominal budget the whole pool was used (Table 4). Paired row-bootstrap intervals for every (dataset, split, method, budget) unit are in the analysis artifact `row_bootstrap.parquet`.

Dataset	Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
Abalone	CatBoost	1.021±0.059	0.852±0.028	0.751±0.024	0.713±0.021	0.675±0.029
	TabPFN v2	0.956±0.082	0.795±0.036	0.702±0.042	0.665±0.019	0.639±0.016
	TabPFN-3	0.958±0.071	0.798±0.029	0.687±0.020	0.639±0.015	0.629±0.014
	Random forest	0.997±0.137	0.850±0.041	0.765±0.052	0.708±0.016	0.678±0.023
	Linear model	1.004±0.227	0.772±0.036	0.729±0.021	0.695±0.006	0.691±0.011
	Constant	1.013±0.020	1.006±0.002	1.003±0.005	1.001±0.000	1.000±0.000
Ames Housing	CatBoost	0.828±0.053	0.607±0.040	0.449±0.012	0.366±0.006	0.325±0.032
	TabPFN v2	0.725±0.113	0.406±0.035	0.339±0.013	0.318±0.013	0.296±0.024

Dataset	Method	$N = 8$	$N = 32$	$N = 128$	$N = 512$	$N = 2{,}048$
	TabPFN-3	0.690±0.075	0.451±0.111	0.400±0.142	0.366±0.166	0.344±0.144
	Random forest	0.836±0.067	0.577±0.035	0.485±0.005	0.411±0.009	0.371±0.045
	Linear model	0.607±0.089	0.578±0.232	0.566±0.193	0.401±0.065	0.361±0.038
	Constant	1.098±0.081	1.009±0.003	1.004±0.004	1.001±0.001	1.002±0.002
Auto MPG	CatBoost	0.519±0.035	0.408±0.047	0.335±0.028	0.320±0.032	–
	TabPFN v2	0.402±0.047	0.414±0.080	0.319±0.013	0.291±0.029	–
	TabPFN-3	0.388±0.041	0.379±0.024	0.318±0.004	0.284±0.014	–
	Random forest	0.498±0.043	0.428±0.052	0.338±0.029	0.331±0.051	–
	Linear model	0.460±0.024	0.456±0.038	0.407±0.023	0.410±0.017	–
	Constant	1.070±0.049	1.018±0.021	1.002±0.002	1.002±0.002	–
Bike Sharing	CatBoost	0.956±0.072	0.865±0.023	0.608±0.060	0.380±0.019	0.288±0.011
	TabPFN v2	0.953±0.072	0.828±0.035	0.541±0.057	0.321±0.017	0.260±0.018
	TabPFN-3	0.983±0.116	0.823±0.022	0.431±0.021	0.326±0.018	0.249±0.023
	Random forest	0.927±0.032	0.813±0.027	0.726±0.036	0.527±0.002	0.360±0.027
	Linear model	1.231±0.411	0.957±0.044	0.828±0.038	0.792±0.023	0.782±0.019
	Constant	1.049±0.072	1.056±0.052	1.003±0.004	1.002±0.001	1.002±0.001
California Housing	CatBoost	0.927±0.101	0.768±0.022	0.596±0.010	0.524±0.023	0.451±0.013
	TabPFN v2	0.912±0.085	0.785±0.160	0.540±0.036	0.493±0.032	0.437±0.018
	TabPFN-3	0.935±0.045	0.774±0.132	0.524±0.028	0.451±0.035	0.400±0.026
	Random forest	1.018±0.199	0.749±0.022	0.604±0.010	0.563±0.014	0.504±0.010
	Linear model	1.733±1.174	0.809±0.081	0.656±0.045	0.643±0.075	0.651±0.031
	Constant	1.085±0.074	1.011±0.018	1.001±0.001	1.002±0.003	1.001±0.002
Concrete Strength	CatBoost	0.933±0.071	0.688±0.100	0.421±0.026	0.293±0.030	0.263±0.025
	TabPFN v2	1.002±0.174	0.523±0.091	0.376±0.026	0.300±0.026	0.261±0.031
	TabPFN-3	1.018±0.161	0.543±0.137	0.364±0.013	0.280±0.037	0.246±0.025
	Random forest	0.979±0.067	0.695±0.107	0.481±0.029	0.370±0.029	0.328±0.033
	Linear model	1.207±0.478	0.738±0.161	0.679±0.069	0.632±0.020	0.628±0.018
	Constant	1.024±0.022	1.017±0.014	1.004±0.003	1.005±0.006	1.006±0.009
Diamonds	CatBoost	0.702±0.225	0.469±0.039	0.326±0.029	0.210±0.025	0.159±0.011
	TabPFN v2	0.631±0.130	0.450±0.157	0.295±0.108	0.161±0.007	0.134±0.006
	TabPFN-3	0.745±0.272	0.497±0.183	0.292±0.126	0.159±0.016	0.130±0.006
	Random forest	0.625±0.204	0.392±0.038	0.364±0.040	0.331±0.039	0.217±0.005
	Linear model	0.600±0.099	0.466±0.131	0.348±0.059	0.304±0.016	0.295±0.027
	Constant	1.054±0.082	1.011±0.010	1.003±0.002	1.001±0.001	1.002±0.001
Insurance Charges	CatBoost	0.751±0.155	0.519±0.029	0.429±0.054	0.410±0.056	0.407±0.060
	TabPFN v2	0.629±0.032	0.518±0.013	0.434±0.071	0.405±0.066	0.403±0.063
	TabPFN-3	0.581±0.056	0.513±0.012	0.404±0.082	0.386±0.073	0.385±0.070
	Random forest	0.668±0.146	0.476±0.063	0.468±0.070	0.429±0.060	0.425±0.061
	Linear model	0.668±0.056	0.567±0.031	0.512±0.049	0.506±0.044	0.504±0.044
	Constant	1.026±0.035	1.006±0.008	1.005±0.006	1.002±0.001	1.001±0.001

Table 11: Crossover category of every (dataset, split) cell against each comparator, LLM arm = gpt-4.1-mini 32-shot. Counts are realised labels, not nominal caps. S(8): comparator already better at the smallest budget; B[N_1, N_2]: first overtaken between the measured budgets N_1 and N_2 (an empirical bracket, not a confidence interval); N(N): not overtaken at any measured budget up to N. Endpoint categories do not establish unobserved crossings outside the grid. A superscript ^{rk} marks k sign changes of the comparison after the first budget at which the comparator led, i.e. the advantage did not persist monotonically.

Dataset	Task	Seed	CatBoost	TabPFN v2	Random forest	Linear model
Abalone	reg	42	S(8)	S(8)	S(8)	S(8)
Abalone	reg	43	B[8, 32]	B[8, 32]	B[8, 32]	B[8, 32]
Abalone	reg	44	B[8, 32]	B[8, 32]	S(8)	S(8)
Adult Income	cls	42	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]
Adult Income	cls	43	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]
Adult Income	cls	44	B[128, 512]	B[128, 512]	B[128, 512]	B[32, 128]
Ames Housing	reg	42	B[128, 512]	B[8, 32]	B[512, 1168]	B[128, 512]
Ames Housing	reg	43	B[128, 512]	B[8, 32]	B[128, 512]	B[512, 1168]
Ames Housing	reg	44	B[128, 512]	B[32, 128]	B[128, 512]	B[512, 1168]
Auto MPG	reg	42	B[8, 32]	S(8)	S(8)	S(8)
Auto MPG	reg	43	S(8)	S(8)	S(8)	S(8)
Auto MPG	reg	44	B[32, 128]	S(8)	B[32, 128]	B[32, 128]^{r1}
Bank Marketing	cls	42	B[8, 32]	B[8, 32]	B[32, 128]	B[8, 32]
Bank Marketing	cls	43	B[8, 32]	B[8, 32]	B[32, 128]	B[32, 128]

Dataset	Task	Seed	CatBoost	TabPFN v2	Random forest	Linear model
Bank Marketing	cls	44	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]
Bike Sharing	reg	42	B[8, 32]	B[8, 32]	B[8, 32]	B[32, 128]
Bike Sharing	reg	43	S(8)	S(8)	S(8)	B[8, 32]
Bike Sharing	reg	44	B[8, 32]	B[8, 32]	B[8, 32]	B[32, 128]
California Housing	reg	42	B[8, 32]	B[32, 128]	B[8, 32]	B[32, 128]
California Housing	reg	43	B[8, 32]	B[8, 32]	B[8, 32]	B[8, 32]
California Housing	reg	44	B[32, 128]	B[8, 32]	B[32, 128]	B[32, 128]
Concrete Strength	reg	42	B[8, 32]	B[8, 32]	B[8, 32]	B[8, 32]
Concrete Strength	reg	43	B[32, 128]	B[8, 32]	B[32, 128]	B[32, 128]
Concrete Strength	reg	44	B[32, 128]	B[8, 32]	B[32, 128]	N(824)
Credit Card Default	cls	42	B[128, 512]	S(8)^{r2}	B[128, 512]	N(2048)
Credit Card Default	cls	43	B[128, 512]	B[8, 32]	B[8, 32]^{r2}	B[512, 2048]
Credit Card Default	cls	44	B[128, 512]	S(8)^{r2}	B[32, 128]	B[128, 512]
Diabetes (Pima)	cls	42	B[128, 512]	B[32, 128]	B[128, 512]	B[32, 128]
Diabetes (Pima)	cls	43	B[512, 614]	B[8, 32]	N(614)	B[8, 32]
Diabetes (Pima)	cls	44	B[128, 512]	B[32, 128]	B[128, 512]	B[32, 128]
Diamonds	reg	42	B[32, 128]	B[32, 128]	B[512, 2048]	B[128, 512]
Diamonds	reg	43	B[32, 128]	B[128, 512]	B[128, 512]	B[128, 512]
Diamonds	reg	44	B[128, 512]	B[8, 32]	B[128, 512]	B[32, 128]
Employee Attrition	cls	42	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]
Employee Attrition	cls	43	B[8, 32]	B[8, 32]	B[32, 128]	B[32, 128]
Employee Attrition	cls	44	B[128, 512]	B[128, 512]	B[32, 128]	B[128, 512]
German Credit	cls	42	B[8, 32]	S(8)^{r2}	S(8)	B[8, 32]
German Credit	cls	43	B[8, 32]	S(8)	S(8)	S(8)
German Credit	cls	44	B[8, 32]	B[8, 32]	S(8)	B[8, 32]
Heart Disease	cls	42	N(242)	N(242)	N(242)	N(242)
Heart Disease	cls	43	N(242)	N(242)	N(242)	N(242)
Heart Disease	cls	44	N(242)	S(8)^{r1}	B[8, 32]^{r1}	N(242)
Insurance Charges	reg	42	B[8, 32]	B[8, 32]	S(8)	B[32, 128]
Insurance Charges	reg	43	B[8, 32]	B[8, 32]	B[8, 32]	B[32, 128]
Insurance Charges	reg	44	B[8, 32]	B[8, 32]	B[8, 32]	B[32, 128]
Online Shoppers	cls	42	B[32, 128]	B[128, 512]	N(2048)	N(2048)
Online Shoppers	cls	43	B[32, 128]	B[32, 128]	B[32, 128]	B[128, 512]^{r1}
Online Shoppers	cls	44	B[128, 512]	B[32, 128]	B[512, 2048]	N(2048)
Telco Churn	cls	42	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]
Telco Churn	cls	43	B[128, 512]	B[128, 512]	B[8, 32]^{r1}	B[512, 2048]
Telco Churn	cls	44	B[128, 512]	B[128, 512]	B[32, 128]	S(8)^{r2}
Wine Quality	cls	42	B[32, 128]	B[32, 128]	B[128, 512]	B[512, 2048]
Wine Quality	cls	43	B[128, 512]	B[128, 512]	B[128, 512]	B[128, 512]
Wine Quality	cls	44	B[32, 128]	B[32, 128]	B[32, 128]	B[32, 128]

Conditional crossover stability. We recompute each crossover using shared row-bootstrap draws for the LLM and all comparator budgets. The artifact `crossover_bootstrap.parquet` records the fraction of draws retaining the original category and the fraction retaining both the category and its realised-label endpoints. The main text reports both medians. These frequencies condition on fitted predictions, exemplars, and the split. They neither include training variability nor form simultaneous confidence intervals across budgets.

E COST, LATENCY, AND COVERAGE

Table 12 reports estimated token cost and measured request latency of the API arms; the recorded responses have an estimated cost of $385.13 under the stated rate assumptions, including the excluded classification arm. The local comparators incur no API cost, and their fit times were not retained in the prediction tables, so we do not report them. Table 13 accounts for every stored prediction: model failures (a timeout, an unparseable reply, or a transport error that survived the retry policy) were 0.0007% of rows, one-class labelled subsets were 2.67% of rows (48 units), and the probability ties discussed in Appendix B are listed per arm.

Table 12: Estimated API token cost and measured latency. Standard uncached input/output rates per million tokens are \$0.40/\$1.60 (gpt-4.1-mini), \$2.50/\$15.00 (gpt-5.4), and \$5.00/\$30.00 (gpt-5.5), checked 17 September 2026 (OpenAI, 2026). Costs reprice recorded usage; they exclude discarded attempts and cache discounts and are not billed totals. Latency is the final recorded request's wall-clock time (24 concurrent requests). The local comparators incur no API cost; their fit times were not retained in the prediction tables and are not reported.

Arm	Shots	Units	Rows	Cost (USD)	USD / 1k rows	Prompt tok. (mean)	Latency s (median)
gpt-4.1-mini	0	54	19,665	2.89	0.147	361	0.60
gpt-4.1-mini	8	54	19,665	10.53	0.536	1,331	0.61
gpt-4.1-mini	32	54	19,665	33.46	1.702	4,247	0.63
gpt-5.4	0	18	6,555	6.36	0.970	360	0.85
gpt-5.4	32	18	6,555	70.08	10.692	4,247	0.88
gpt-5.5	0	18	6,555	68.63	10.471	369	5.44
gpt-5.5	32	18	6,555	193.17	29.470	4,256	4.77

Table 13: Coverage and failure accounting over all stored predictions. "one_class" is a recorded protocol condition (a label-blind subset with one class), not a model failure. The lower panel counts classification rows whose extracted probability is exactly 0.5, a diagnostic of probability ties rather than a uniquely identified cause. Single-entry top-k output can create these ties; exact equal mass or a fallback can too. If neither class token is returned, extraction fails and invokes the retry/fallback policy.

Status	Rows	Share
ok	654,682	97.3321%
one_class	17,940	2.6672%
failed_TimeoutError	3	0.0004%
unparseable	1	0.0001%
failed_URLError	1	0.0001%
Classification rows at exactly $P(y{=}1) = 0.5$, by logprob arm		
gpt-4.1-mini 0-shot	62	0.56%
gpt-4.1-mini 32-shot	33	0.30%
gpt-4.1-mini 8-shot	48	0.43%
gpt-5.4 0-shot	3,378	91.08%
gpt-5.4 32-shot	3,443	92.83%

F HISTORICAL FITTED-CURVE SENSITIVITY

The historical cohort of Section 6 located crossovers by fitting a power law $e(N) = aN^{-b} + c$ to each pooled classical learning curve and solving $e(N^\star) = e_{\mathrm{LLM}}$ against a flat LLM error. Table 14 lists the 80 (dataset, classical model) fits, each dataset's post-hoc best LLM configuration (an oracle chosen on evaluation data, hence LLM-favourable), the implied $N^\star$, and the outcome category. 10 fits fail the cohort's reliability check and are excluded from its headline counts; the median fitted exponent among the rest is 0.476.

For a valid fit ($a, b > 0$ and $e_{\mathrm{LLM}} > c$), $N^\star = \big(a/(e_{\mathrm{LLM}} - c)\big)^{1/b}$ and

$$\frac{\partial \log N^\star}{\partial e_{\mathrm{LLM}}} \;=\; -\frac{1}{b\,(e_{\mathrm{LLM}} - c)}. \tag{1}$$

The sensitivity is large when the exponent b is small (a flat curve) or when the LLM error sits close to the fitted floor c. Among the 35 reliable cells whose crossover lies inside or beyond the measured range, the median absolute sensitivity is 16.6 and the largest is 461.2 per unit of error: these derivatives describe infinitesimal perturbations. Multiplying them by a finite error change and exponentiating is only a local approximation, which can fail near the fitted floor or across outcome categories. This is why the main text reports measured brackets and treats fitted crossovers as a secondary analysis: the sensitivity analysis shows how fragile an extrapolated $N^\star$ is; it does not validate the extrapolation. Of the 70 reliable cells, 10 have no crossing within the data range under the

fitted curve, of which 6 lie below the fitted floor; an LLM error below a fitted comparator asymptote is a property of the extrapolation, not an empirical demonstration of permanent superiority.

Table 14: Historical cohort: pooled classical learning-curve fits $e(N) = aN^{-b} + c$ from the companion classical-scaling project, each dataset's post-hoc best LLM configuration (an oracle selected on evaluation data, hence LLM-favourable), the implied crossover $N^\star$, and the conditional sensitivity $\partial \log N^\star / \partial e_{\mathrm{LLM}} = -1/[b\,(e_{\mathrm{LLM}} - c)]$ for fits with $a, b > 0$ and $e_{\mathrm{LLM}} > c$. A dagger marks fits that fail the cohort's reliability check; $N^\star$ is blank where the LLM error lies below the fitted floor ($e_{\mathrm{LLM}} \leq c$, "never" overtaken by the extrapolation) or the fit is degenerate. $N^\star$ is given relative to the full training split N_{full}. Outcomes are fit-implied, not observed brackets: min = fitted comparator better at the smallest size; within = fitted crossing inside the range; beyond = $N^\star > N_{\mathrm{full}}$; never = below the fitted floor; extreme = $N^\star$ more than $1000 \times N_{\mathrm{full}}$.

Dataset	Model	a	b	c	R^2	e_{LLM}	$N^\star/N_{\mathrm{full}}$	Sens.	Outcome
Abalone	Boosting	0.635	0.280	0.612	0.97	1.177	4.56×10^{-4}	-6.34	min
Abalone	Lasso†	162	3.000	0.993	0.00	1.177	2.87×10^{-3}	-1.81	min
Abalone	RandomForest	0.809	0.431	0.657	0.98	1.177	8.38×10^{-4}	-4.47	min
Abalone	Ridge	0.917	0.721	0.668	0.97	1.177	6.77×10^{-4}	-2.72	min
Abalone	SVM	0.736	0.462	0.653	0.97	1.177	6.25×10^{-4}	-4.13	min
Adult Income	Boosting	2.67	0.679	0.079	0.98	0.180	3.43×10^{-3}	-14.5	within
Adult Income	LinearModel	1.5	0.556	0.091	0.98	0.180	4.46×10^{-3}	-20.2	within
Adult Income	RandomForest	1.72	0.695	0.099	0.94	0.180	2.25×10^{-3}	-17.8	min
Adult Income	SVM	0.708	0.438	0.102	0.87	0.180	4.30×10^{-3}	-29.4	within
Ames Housing	Boosting	2.33	0.595	0.298	0.95	0.392	0.189	-17.9	within
Ames Housing	Lasso	0.0307	0.511	0.999	0.69	0.392	–	–	never
Ames Housing	RandomForest	1.06	0.164	0.000	0.96	0.392	0.366	-15.6	within
Ames Housing	Ridge†	382	3.000	0.500	0.01	0.392	–	–	never
Ames Housing	SVM	1.63	0.392	0.423	1.00	0.392	–	–	never
Auto MPG	Boosting	0.593	0.098	0.000	0.71	0.466	0.0363	-21.9	min
Auto MPG	Lasso†	4.83×10^{-8}	0.003	1.005	-0.00	0.466	–	–	never
Auto MPG	RandomForest	0.539	0.095	0.000	0.75	0.466	0.0143	-22.5	min
Auto MPG	Ridge	1.56×10^{3}	3.000	0.395	0.74	0.466	0.0877	-4.64	min
Auto MPG	SVM	5.43×10^{3}	3.000	0.310	0.92	0.466	0.103	-2.13	within
Bank Marketing	Boosting	9.94	0.820	0.074	1.00	0.369	2.02×10^{-3}	-4.13	min
Bank Marketing	LinearModel	43	1.051	0.095	1.00	0.369	3.40×10^{-3}	-3.47	within
Bank Marketing	RandomForest	1.25	0.473	0.063	1.00	0.369	5.36×10^{-4}	-6.9	min
Bank Marketing	SVM	0.463	0.274	0.067	0.93	0.369	1.30×10^{-4}	-12	min
Bike Sharing	Boosting	20.5	0.965	0.388	0.97	0.641	6.83×10^{-3}	-4.09	min
Bike Sharing	Lasso	0.507	0.932	1.000	0.93	0.641	–	–	never
Bike Sharing	RandomForest	2.94	0.348	0.128	1.00	0.641	0.0109	-5.61	within
Bike Sharing	Ridge	155	1.764	0.776	0.95	0.641	–	–	never
Bike Sharing	SVM	0.976	0.046	0.000	0.98	0.641	0.628	-33.7	within
California Housing	Boosting	6.1	0.677	0.467	1.00	0.750	5.64×10^{-3}	-5.22	min
California Housing	Lasso	0.969	0.000	0.036	0.70	0.750	2.07×10^{267}	-2.87×10^{3}	extreme
California Housing	RandomForest	0.937	0.076	0.000	0.99	0.750	1.14×10^{-3}	-17.6	min
California Housing	Ridge†	0.789	0.020	0.000	0.46	0.750	7.80×10^{-4}	-67.2	min
California Housing	SVM	1.49	0.443	0.509	0.99	0.750	3.70×10^{-3}	-9.38	min
Concrete Strength	Boosting	3.94	0.719	0.307	0.97	0.737	0.0265	-3.24	min
Concrete Strength	Lasso†	0.024	0.033	0.971	0.00	0.737	–	–	never
Concrete Strength	RandomForest	1.24	0.328	0.205	0.98	0.737	0.0162	-5.74	min
Concrete Strength	Ridge†	0.651	0.009	0.000	0.33	0.737	1.93×10^{-9}	-147	min
Concrete Strength	SVM	1.41	0.511	0.377	0.98	0.737	0.0176	-5.45	min
Credit Card Default	Boosting	1.4	0.466	0.210	0.98	0.424	2.35×10^{-3}	-10	min
Credit Card Default	LinearModel	0.173	0.335	0.285	0.78	0.424	7.90×10^{-5}	-21.4	min
Credit Card Default	RandomForest	0.496	0.415	0.238	0.90	0.424	4.42×10^{-4}	-12.9	min
Credit Card Default	SVM†	0.23	0.216	0.258	0.25	0.424	1.87×10^{-4}	-27.9	min
Diabetes (Pima)	Boosting	0.692	0.213	0.000	0.83	0.283	0.108	-16.6	within
Diabetes (Pima)	LinearModel	125	2.038	0.180	0.95	0.283	0.053	-4.76	within
Diabetes (Pima)	RandomForest	0.472	0.147	0.000	0.95	0.283	0.052	-24	within
Diabetes (Pima)	SVM	0.707	0.195	0.000	0.81	0.283	0.176	-18.1	within
Diamonds	Boosting	8.67	0.755	0.208	0.93	0.402	3.57×10^{-3}	-6.85	within
Diamonds	Lasso†	1.00×10^{-8}	2.355	0.993	-0.00	0.402	–	–	never
Diamonds	RandomForest	1.5	0.326	0.112	0.98	0.402	3.57×10^{-3}	-10.6	within
Diamonds	Ridge	305	1.587	0.285	0.98	0.402	3.30×10^{-3}	-5.39	within
Diamonds	SVM	1.28	0.290	0.104	0.59	0.402	3.52×10^{-3}	-11.5	within
Employee Attrition	Boosting	1.65	0.473	0.153	0.97	0.198	1.74	-47	beyond
Employee Attrition	LinearModel	3.39	0.647	0.139	0.94	0.198	0.452	-26.4	within
Employee Attrition	RandomForest	3.39	0.723	0.214	0.96	0.198	–	–	never
Employee Attrition	SVM	2.1	0.404	0.043	0.74	0.198	0.541	-16	within
German Credit	Boosting	0.772	0.479	0.207	0.95	0.474	0.0115	-7.84	min
German Credit	LinearModel†	0.343	0.038	0.000	0.40	0.474	2.59×10^{-7}	-55.5	min
German Credit	RandomForest	0.396	0.229	0.131	0.89	0.474	2.34×10^{-3}	-12.7	min
German Credit	SVM	1.02×10^{4}	3.000	0.248	0.90	0.474	0.0445	-1.48	within
Heart Disease	Boosting	17.1	1.409	0.039	0.98	0.132	0.167	-7.62	within
Heart Disease	LinearModel	0.735	0.454	0.000	0.92	0.132	0.182	-16.7	within

Dataset	Model	a	b	c	R^2	e_{LLM}	$N^\star/N_{\mathrm{full}}$	Sens.	Outcome
Heart Disease	RandomForest	0.126	0.161	0.000	0.77	0.132	3.12×10^{-3}	-47	min
Heart Disease	SVM	0.379	0.504	0.028	0.98	0.132	0.0542	-19.1	min
Insurance Charges	Boosting	0.818	0.349	0.278	0.95	0.465	0.0642	-15.3	within
Insurance Charges	Lasso†	0.752	0.001	0.254	0.02	0.465	–	-4.76×10^{3}	extreme
Insurance Charges	RandomForest	1.64	0.722	0.369	0.86	0.465	0.0482	-14.5	within
Insurance Charges	Ridge	2.68	1.050	0.463	0.95	0.465	0.815	-436	within
Insurance Charges	SVM	2.69	0.631	0.336	0.97	0.465	0.115	-12.3	within
Online Shoppers	Boosting	1.16	0.517	0.062	1.00	0.114	0.0419	-37.4	within
Online Shoppers	LinearModel	7.62	0.844	0.111	0.99	0.114	1.31	-461	beyond
Online Shoppers	RandomForest	0.553	0.416	0.071	1.00	0.114	0.0491	-56.9	within
Online Shoppers	SVM	0.32	0.251	0.099	0.94	0.114	21.6	-271	beyond
Telco Churn	Boosting	0.737	0.417	0.137	0.99	0.176	0.205	-61.6	within
Telco Churn	LinearModel	4.03	0.867	0.157	1.00	0.176	0.0873	-61.7	within
Telco Churn	RandomForest	0.707	0.722	0.173	0.97	0.176	0.32	-441	within
Telco Churn	SVM	2.10×10^{3}	2.718	0.201	0.90	0.176	–	–	never
Wine Quality	Boosting	0.629	0.277	0.079	1.00	0.338	4.79×10^{-3}	-14	min
Wine Quality	LinearModel	1.97	0.839	0.195	0.96	0.338	4.43×10^{-3}	-8.4	min
Wine Quality	RandomForest	0.528	0.194	0.000	0.96	0.338	1.94×10^{-3}	-15.3	min
Wine Quality	SVM	348	1.652	0.170	0.95	0.338	0.0196	-3.61	within

G Replication cohort: recomputed crossovers and implementer-block bootstrap

Each of the 126 implementations reported its own LLM error per (dataset, configuration). To ask whether implementations disagree about the qualitative outcome rather than merely about a number, we recompute every implementation's crossover against the same pooled classical curve of Appendix F, so that implementations differ only in the LLM error they measured. Table 15 summarises the result over the 560 (dataset, classical model, configuration) cells with a reliable fit and at least two implementations. In 60.7% of cells at least one implementation lands in a different category from the others, although the majority category is stable: the median share of implementations in a cell's majority category is 96.6%, and 7.0% of implementation–comparison records (implementer-block bootstrap 95% interval 4.7–9.5%) reach a different category from the one implied by their cell's median error.

Uncertainty for the cohort statistics resamples implementer identities with replacement (2000 draws), carrying every dataset an implementer covered. This preserves within-implementer dependence, conditional on the fixed dataset collection and fitted reference curves; it does not propagate uncertainty in those curves or establish independence between implementers. Under that scheme the median coefficient of variation of LLM error, 0.1476, has a 95% interval of 0.0808–0.1704. The share of cells with any disagreement is reported without an interval because it depends on cohort size by construction: a resample that duplicates implementers cannot produce disagreement between the duplicates, so a bootstrap of that quantity is biased downward. The supplied starter uses hard-label AUROC for classification and scores parsed rows only (Table 3). It supplies shared data and API helpers, so these are not independent pipeline designs. The historical protocol also differs from the controlled benchmark in several respects (an evaluation sample of at most 100 rows per dataset, implementer-chosen preprocessing, gpt-4.1-nano and gpt-4.1-mini at zero and five shots, two prompt formats), and the numbers in this section describe variation under that protocol only. Variation combines evaluation sampling, API outputs, and implementation choices; it is not an isolated causal effect of coding decisions. The historical preprocessing can also use full-table imputation, unlike the controlled benchmark.

Table 15: Historical cohort: agreement across separately produced implementations about the qualitative crossover category, with every implementation's crossover recomputed against the same pooled fitted classical curve. A cell is a (dataset, classical model, LLM configuration) with a reliable fit and at least two implementations. "Disagree" is the share of cells in which at least one implementation lands in a different category from the others; "majority share" is the median fraction of implementations in the cell's majority category; "off-median" is the share of implementation–comparison records whose category differs from that of the cell's median LLM error.

Dataset	Cells	Impl./cell (median)	Disagree	Majority share	Off-median
Abalone	32	29	28%	100%	1.0%
Adult Income	32	18	25%	100%	5.9%
Ames Housing	32	29	94%	93%	8.7%
Auto MPG	32	28	22%	100%	1.9%
Bank Marketing	32	19	31%	100%	2.5%
Bike Sharing	40	18	38%	100%	3.8%
California Housing	32	18	97%	89%	11.8%
Concrete Strength	24	29	100%	97%	3.4%
Credit Card Default	24	18	96%	92%	10.2%
Diabetes (Pima)	32	18	100%	83%	22.0%
Diamonds	32	18	50%	97%	5.6%
Employee Attrition	32	18	56%	94%	7.5%
German Credit	24	18	100%	83%	13.2%
Heart Disease	32	18	25%	100%	4.7%
Insurance Charges	32	29	100%	97%	9.3%
Online Shoppers	32	17	19%	100%	2.4%
Telco Churn	32	19	100%	95%	16.6%
Wine Quality	32	17	47%	100%	3.1%
All	560	18	61%	97%	7.0%

H INPUT-INFORMATION ABLATIONS

The ablations were run in the historical pipeline, not in the controlled benchmark: gpt-4.1-nano, zero-shot, key–value serialisation, on 6 classification datasets (3 of high and 3 of low prior ubiquity) with the same fixed evaluation rows per dataset (at most 100; the whole held-out split where smaller). One factor is moved at a time relative to the original prompt: feature names replaced by neutral tokens with the task description kept; a generic target description with real feature names kept; and numeric values replaced by labelled deciles ("d (0=low . . . 9=high)") with names kept. A logistic regression trained on the full training split and scored on the same rows, with original versus decile-binned features, serves as a trained positive control.

Table 16 gives the per-dataset changes. Using unadjusted per-dataset intervals, replacing values by deciles hurts the frozen model on 3 of the 6 datasets, all in the high-ubiquity tier, and the high−low contrast is +0.246 (95% interval +0.143 to +0.346); the trained control loses at most 0.094 AUROC under the same binning, so the binned features retain the structure a trained model uses. Anonymising feature names or the task description has weaker and inconsistent effects (2 and 1 datasets with unadjusted intervals excluding zero; contrasts -0.005 and +0.026, both with intervals covering zero). These exploratory comparisons were not adjusted for multiple testing. They show sensitivity to the representation and information in numeric inputs. We do not read it as a measurement of memorisation: the high-ubiquity datasets are also the ones on which the model beats chance, so the contrast is confounded with baseline accuracy, and binning changes both the information and the representation available to the model. The interventions probe sensitivity to input information; they do not identify a mechanism.

Table 16: Input-information ablations (historical pipeline; gpt-4.1-nano, zero-shot, key–value serialisation, six classification datasets, the same fixed evaluation rows per dataset – at most 100, or the whole held-out split where smaller). Entries are the change in error ($1 - \mathrm{AUROC}$) relative to the original prompt, with a paired row-bootstrap 95% interval; positive means the intervention hurt. *feat*: feature names replaced by neutral tokens, task description kept; *task*: generic target wording, real feature names kept; *value*: numeric values replaced by labelled deciles, names kept. The positive control is a logistic regression trained on the full training split and scored on the same rows with original versus decile-binned features. "Tier" is the dataset's prior-ubiquity group used to form the high−low contrast.

Dataset	Tier	Δ feat	Δ task	Δ value	Control AUROC (orig)	Control AUROC (binned)
Adult Income	high	-0.067 [-0.169, +0.034]	+0.180 [+0.091, +0.276]	+0.180 [+0.091, +0.276]	0.935	0.928
Diabetes (Pima)	high	+0.050 [-0.072, +0.180]	+0.008 [-0.052, +0.064]	+0.111 [+0.012, +0.202]	0.817	0.836
Heart Disease	high	+0.183 [+0.054, +0.306]	+0.011 [-0.096, +0.112]	+0.361 [+0.165, +0.541]	0.951	0.857
Employee Attrition	low	+0.037 [-0.100, +0.167]	+0.088 [-0.005, +0.194]	-0.034 [-0.188, +0.124]	0.888	0.885
Online Shoppers	low	+0.088 [+0.046, +0.131]	+0.025 [-0.098, +0.140]	-0.047 [-0.191, +0.087]	0.968	0.956
German Credit	low	+0.067 [-0.003, +0.128]	+0.007 [+0.000, +0.023]	+0.000 [+0.000, +0.000]	0.693	0.685
High−low contrast		-0.005 [-0.089, +0.083]	+0.026 [-0.049, +0.093]	+0.246 [+0.143, +0.346]		

I Execution settings and reproduction limits

Classical and foundation models. CatBoost uses its native categorical interface, `random_seed` equal to the split seed, and otherwise library defaults. Random forests use the split seed and `n_jobs=-1`; the linear arm is logistic regression with `max_iter=2000` for classification and Ridge with default regularisation for regression. Imputation, one-hot encoding, and (for the linear arm) scaling are fitted only to the budgeted subset. TabPFN uses `device=mps`, `inference_precision=auto`, the split seed, `ignore_pretraining_limits=True`, and prediction chunks of 100 rows. Checkpoints are `tabpfn-v2-classifier-finetuned-zk73skhh.ckpt`, `tabpfn-v2-regressor.ckpt`, `tabpfn-v3-classifier-v3_default.ckpt`, and `tabpfn-v3-regressor-v3_default.ckpt`. The original label-range guard permits up to two additional executions after a scale flag. Its final flag was not preserved by the runner, and the guard does not certify numerical accuracy. The revised runner records the failure and uses an allowed-label empirical fallback after exhausted retries.

LLM requests. The harness uses Chat Completions with recorded aliases `gpt-4.1-mini`, `gpt-5.4`, and `gpt-5.5`. No temperature or sampling seed is specified. The primary model allows one output token for classification and 16 for regression. gpt-5.4 uses `max_completion_tokens=2048`; gpt-5.5 uses 4096 and `reasoning_effort=low`. The latter settings were chosen after empty responses in feasibility runs. There are 24 concurrent requests per LLM unit. Model aliases and output-token settings are not a guarantee of repeatability.

Saved outputs versus fresh executions. Controlled metrics and bootstrap summaries can be recomputed from per-row predictions; historical summaries additionally require the cohort aggregates and fitted reference curves. The incident comparison requires the retained suspect predictions. Raw dataset hashes and frozen split identifiers are stored in the manifest. Package versions, checkpoint hashes, returned API snapshot identifiers, and system fingerprints were not recorded in that manifest, so exact fresh execution cannot be certified from it. The stored predictions preserve the results that were actually analysed; they do not replace complete execution provenance. TabPFN checkpoints require separate access under their applicable licences and are not part of the prediction artifacts.

J Repeated-record sensitivity

We hash each raw feature vector without its positional index and count matches in both the full training pool and the largest labelled subset actually used. The latter contains every smaller nested subset and all LLM exemplars. Target equality is checked separately. Counts in Table 17 concern exact stored feature values; they do not identify real-world entities or detect approximate duplicates.

The sensitivity removes every evaluation row matching features in the largest used training subset, applying the same mask to all methods and budgets. Predictions are not refitted. Regression nor-

Table 17: Evaluation rows matching training features, shown as counts for seeds $42/43/44$. Full pool includes unused training rows; used subset is the largest labelled subset actually evaluated. The last column additionally requires target equality. Other datasets have no feature matches to used training rows. Repeated features alone do not establish real-world entity identity.

Dataset	Full pool: X	Used: X	Used: X and y
adult income	2/2/0	1/0/0	1/0/0
concrete strength	11/11/13	11/11/13	6/6/9
diamonds	3/3/7	0/1/0	0/0/0
insurance charges	1/0/0	1/0/0	1/0/0
online shoppers	3/8/5	1/4/1	1/4/1
telco churn	6/4/7	2/2/3	1/1/2
wine quality	146/146/135	74/65/57	74/65/57

malization is recomputed from the remaining evaluation targets. Against GPT-4.1-mini, primary categories for CatBoost and both TabPFN generations are unchanged; CatBoost has two location changes and TabPFN v3 has three. Against GPT-5.5 on Wine Quality, CatBoost and TabPFN v3 move from $(128, 512]$ to $(512, 2048]$, while TabPFN v2 becomes not overtaken in range. These are exploratory rescored outcomes, conditional on existing predictions. The complete row counts and before/after endpoints are provided with the saved analysis outputs.

K Foundation-model generation contrast

Both generations were evaluated on identical rows. Averaging budgets and splits within each dataset before averaging datasets, mean error is 0.2113 (v2) versus 0.2125 (v3) on classification and 0.5053 versus 0.5029 on regression. The paired dataset-level differences (v3 minus v2) are +0.0012 [-0.0033, +0.0055] and -0.0025 [-0.0156, +0.0127], respectively (95% dataset-block intervals). Neither two-sided dataset-level signed-rank test separates the releases ($p = 0.625$ and $p = 0.742$). Although v3 wins 66.7% of regression budget–split cells, those correlated cells do not establish a significant aggregate improvement. These contrasts concern the evaluated defaults and dataset collection.

L Hypothetical LLM error reduction

For dataset–split cell d, comparator m, and measured budget N we define the error reduction an LLM would need to match that comparator:

$$H_{d,m}(N) = \max\big(0,\ e_{\mathrm{LLM},d} - e_{m,d}(N)\big). \tag{2}$$

Zero headroom means the LLM already matches or beats the comparator. Median headroom over dataset–split–budget cells against CatBoost is 0.0023 on classification (49.7% of cells at zero) and 0.1007 on regression (29.9% at zero). There is no common absolute-error unit across $1 - \mathrm{AUROC}$ and normalised RMSE, so these distributions are reported by task type and never pooled.

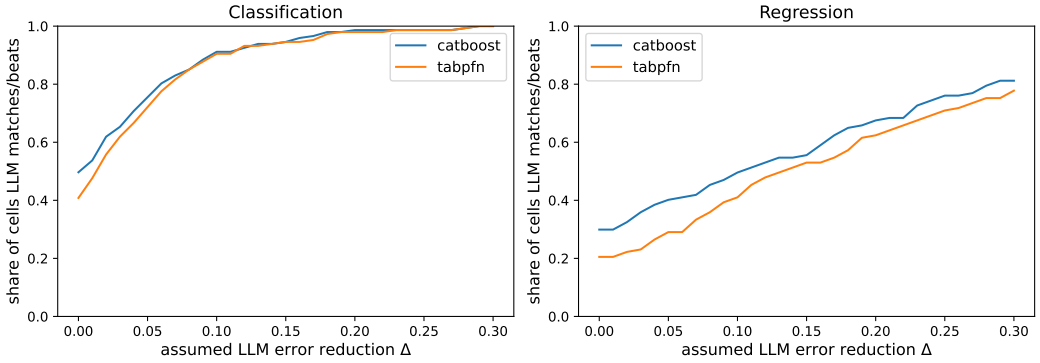


Figure 3: Hypothetical improvement sweep: the share of cells an LLM would match or beat if its error fell by Δ, with $e'_{\mathrm{LLM}} = \max(0, e_{\mathrm{LLM}} - \Delta)$. This is a sensitivity scenario, not a prediction about future models, and Δ is not a measured generational step.

Figure 3 sweeps a hypothetical uniform improvement. We stress what it is not: Δ is an assumption, not an observed capability gain, and the curve says nothing about whether any particular future model will realise it.

M RETAINED EXECUTION DISCREPANCY

An early TabPFN v2 execution, recorded as concurrent with other accelerator jobs, produced regression error more than twice the accepted run's in 14 of 117 units, and more than ten times in 5. The worst median prediction was 50 times the median target. No exception or failure status identified these units. Classification probabilities differed by at most 0.050 (AUROC by 0.0051). The retained suspect and accepted outputs establish the discrepancy: their mean regression errors are 1.475 and 0.509, respectively. Run notes attribute the accepted execution to an unloaded machine, but no second accepted repetition or controlled load experiment was retained; the precise cause and reproducibility remain unverified. The allowed-label range check is only a coarse backstop and would miss the worst suspect unit. This incident motivates retaining and comparing outputs across executions; it does not establish model determinism or isolate a hardware cause.

A subsequent code audit found that the original runner also discarded a returned `implausible_scale` flag after the allowed retries. The revised runner preserves that flag and uses the empirical prediction from the allowed training labels as a declared fallback. This forward correction does not retroactively validate or alter the accepted or suspect predictions.