MSBoost: Using Model Selection with Multiple Base Estimators for Gradient Boosting

Agnij Moitra
Amity International School (Saket)
agnijmoitra[at]outlook[dot]com

Abstract

Gradient boosting is a widely used machine learning algorithm for tabular regression, classification and ranking. Although, most of the open source implementations of gradient boosting such as XGBoost, LightGBM and others have used decision trees as the sole base estimator for gradient boosting. This paper, for the first time, takes an alternative path of not just relying on a static base estimator (usually decision tree), and rather trains a list of models in parallel on the residual errors of the previous layer and then selects the model with the least validation error as the base estimator for a particular layer. This paper has achieved *state-of-the-art* results when compared to other gradient boosting implementations on 50+ tabular regression and classification datasets. Furthermore, ablation studies show that MSBoost is particularly effective for small and noisy datasets. Thereby, it has a significant social impact especially in tabular machine learning problems in the domains where it is not feasible to obtain large high quality datasets.

1 Introduction

Gradient boosting [1, 2] has been a powerful boosting [3] based machine learning algorithm that has achieved state-of-the-art accuracy in various real world tasks. Such as in particle physics, biochemistry, finance, fraud detection, search engine recommendations, drug discovery and many others [4–16]. Its significance lies in its ability to handle diverse data types and complex feature engineering whilst effectively managing high-dimensional, noisy datasets with heterogeneous features.

It builds a 'stronger' predictive model by combining several weaker models through an iterative greedy process that focuses on correcting the errors of previous models, which is based on sound theoretical evidence as per [17]. Popular implementations of gradient boosting include XGBoost [18], which enhances traditional methods by introducing regularization to prevent overfitting and tree pruning to improve efficiency, and LightGBM [19], which differs by using a leaf-wise tree growth strategy instead of level-wise growth, and implements Gradient-based One-Side Sampling (GOSS) to speed up training on large datasets while maintaining accuracy. Furthermore, other variants include CatBoost [20] which introduces a novel categorical encoding method to mitigate target leakage, and using Artificial Neural Network, Principal Component Analysis and Random Projections for feature extraction and combine this with gradient boosting as per AugBoost [21].

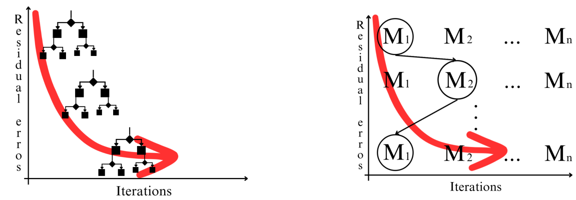
The main contribution of this paper, Model Selection based Gradient Boosting (MSBoost[1]), is to explore, for the first time, the usage of model selection in order to find the base estimator with the least validation error. Unlike the current methods which use a single base estimator, usually decision tree [22–24], although previous research has been done in boosting other models [25]. Benchmarking this method, MSBoost, on 50+ datasets indicate that this method outperforms previous methods such as LightGBM and XGBoost, and based on the ablation studies performed it can be observed that

[1] https://github.com/Agnij-Moitra/MSBoost

Preprint. Under review.

MSBoost is particularly effective for small and noisy datasets. Thereby, MSBoost would particularly effective for tabular regression and classification problems where it is not feasible or expensive to obtain thousands of high quality samples.

2 Method



(a) Conventional Gradient Boosting

(b) Model Selection based Gradient Boosting (MS-Boost)

Figure 1: Conventional Gradient Boosting methods usually use Decision Trees, also known as CART(s), as the sole base estimator in order to minimize the residual errors over a number of iterations. Whereas, MSBoost from a list of ML models dynamically would choose the one with the least residual errors, in parallel, and use it as the base estimator for that layer.

Similar to gradient boosting, the goal of MSBoost is to approximate any arbitrary but particular $\mathcal{F} : \mathbb{R}^m \to \mathbb{R}$ with a series of additive and scaled F_i in order to minimize $\mathcal{L}(\mathcal{F}(\mathbf{x}), F(\mathbf{x}))$. For any given tabular dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, and a differentiable loss function $\mathcal{L}(\mathbf{y}, F(\mathbf{x}))$. Wherein $\mathbf{x}_i$ is an arbitrary but particular vector $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^m)$ containing m features, and $\mathbf{y} \in \mathbb{R}^n$, which has n samples is the target vector. First, MSBoost initializes the first estimator as a constant term i.e $F_0(\mathbf{x}) = \arg\min_k \sum_{i=1}^n \mathcal{L}(y_i, k)$, which turns out to be the arithmetic mean of the target values vector $\mathbf{y}$. Next, for each subsequent iteration $i = 1, \ldots, N$ it shall compute the pseudo residuals:

$$\mathbf{r}_i = -\left[\frac{\partial \mathcal{L}(\mathbf{y}, F_{i-1}(\mathbf{x}))}{\partial F_{i-1}(\mathbf{x})}\right] \tag{1}$$

and the base estimator for i^{th} layer is based on a list of models $\mathcal{M}$, such that:

$$h_i(\mathbf{x}) = \arg\min_{\forall M \in \mathcal{M}} \mathcal{L}(\mathbf{y}, M(\mathbf{r}_i))(\mathbf{r}_i) \tag{2}$$

Finally it would update the model for i^{th} layer, i.e $F_i(\mathbf{x}) = F_{i-1}(\mathbf{x}) + \alpha \cdot h_i(\mathbf{x})$, and the final prediction, $\hat{\mathbf{y}} = F(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{i=1}^N F_i(\mathbf{x})$.

2.1 Rationale for Model Selection in Gradient Boosting

Since model selection searches for $\arg\min_{M \in \mathcal{M}} \mathcal{L}(\mathbf{y}, M(\mathbf{r}_i))$ for each iteration i, $\Rightarrow$ $\mathcal{L}(\mathbf{y}, M(\mathbf{r}_i)) \leq \mathcal{L}(\mathbf{y}, S(\mathbf{r}_i)), \forall S$ which are static machine learning models say Decision Tree. And over a large number of iterations N, $\mathbf{r}_{i,\mathcal{M}}$ (model selection, a dynamic method) $<$ $\mathbf{r}_{i,S}$ (for any static base estimator). This is technically a "$\leq$" inequality, but based on the inductive proposition that over a large number of iterations, N, a static method would have higher $\mathbb{E}(\mathbf{r}_i)$ than dynamically selecting base estimators in each iteration, the "$<$" inequality should hold true. Wherein the base case is $\mathbb{E}(\mathbf{r}_{i,\mathcal{M}})$ < $\mathbb{E}(\mathbf{r}_{i,S})$, which is empirically true as per [26, 27] and theoretically justified by the*No Free Lunch Theorem* [28, 29]. Furthermore, analysing the effect of specific base estimators stacked over N iteration on the residual plots shall be an interesting obsevation, for example a non-linear model like [30] may have a more linear residual plot when compared to that of a linear model, so a non linear base estimator in i^{th} iteration may lead to a linear model in $i+1^{th}$ iteration. But this has been left for a avenue for future research.

Also, as empirically demonstrated by [26, 27], there is no *one-size-fits-all* baseline model which does well on all types of datasets, which empirically justifies as to why boosting multiple estimators might be effective; and, increase the diversity of the base learners, which potentially help to improve the generalization performance (i.e less variance) [31].

2.2 Model Selection Methods[2]

Naïve Method The naïve way for model selection is to train all the available base estimators on $\mathbf{r}_i$ in parallel. This way would ensure that the model with the least residual errors is truly being selected for each layer and precisely conforms to the theoretical rationale stated in Section 2.1. But this would have the largest time complexity, i.e $O(\text{Number of Iterations} \times$ Base model with the highest time complexity i.e the limiting factor).

Random Sampling Sampling a subset of models from $\mathcal{M}$, shall reduce the overall training time, but this may not be find the model with least possible validation residual errors.

Frequency & Probability Based Sampling Assuming that only a subset of models from $\mathcal{M}$ would be used for most of the time due to the characteristics of the dataset being used. For the first I iterations, this shall be a track of the frequency of the top N models, and for the rest of the iterations only train the top N models initially found. Here I and N are hyper-parameters. A more vigorous method for this would be to use Bayesian model selection [32–34] and train the models with the top N probabilities of being used.

3 Experiments & Discussion

Comparison with baselines MSBoost (random sampling half of the models from $\mathcal{M}$ for training in each iteration) was compared[3] with XGBoost and LightGBM. The source code of the experiments are available, and can be reproduced (`https://github.com/Agnij-Moitra/MSBoost`). Unfortunately due to constrained computational resources the benchmarking was done on 1K samples on OpenML [35, 36] datasets with 0.01 lasso threshold to screen for irrelevant features which would have increased the computational costs. Table 1 compares the mean squared error with 5 fold cross validation (CV), and Table 2 compares the log loss with 5 Fold CV; please check Appendix B.1 for entire results. Paired single tailed t-test reveal that MSBoost yields a statistically significant improvement over LightGBM and XGBoost in metrics, with p-value $<< 0.001$ (excluding outliers like wave_energy), and $p < 0.02$ for standard deviation thus improving the bias-variance trade-off [37]. It should be noted even without regularization, and GOSS and EFB of XGBoost and LightGBM respectively, MSBoost has a statistically significant improvement. Thereby, this may have even better improvement over previous methods if those techniques are incorporated in MSBoost.

Table 1: Comparison (regression) with baselines based on mean squared error (MSE)

	MSBoost	LightGBM	XGBoost
wave_energy	0.0 ± 0.0	1.9e+9 ± 2.9e+8	3.0e+9 ± 4.5e+8
Friedman 2	150 ± 31	385 ± 57	501 ± 58
Sparse Uncorr.	1.0 ± 0.15	1.5 ± 0.11	1.7 ± 0.22
kin8nm	2.1e-2 ± 1e-4	3.1e-2 ± 1.5e-3	3.6e-2 ± 1.3e-3
sarcos	32 ± 8	46 ± 15	48 ± 10
Moneyball	431 ± 24	588 ± 42	635 ± 39
yprop_4_1	7e-4 ± 1e-4	9e-4 ± 1e-4	1.1e-3 ± 1e-4
fps_benchmark	2354 ± 110	2917 ± 104	3758 ± 395
Zurich Transport	10 ± 0.7	12 ± 0.9	15 ± 1.4
Diabetes	3017 ± 333	3590 ± 433	3991 ± 651

Table 2: Comparison (classification) with baselines based on log loss

	MSBoost	LightGBM	XGBoost
phoneme	0.34 ± 0.03	0.43 ± 0.07	0.43 ± 0.06
guillermo	0.56 ± 0.04	0.69 ± 0.10	0.77 ± 0.11
MagicTelescope	0.40 ± 0.04	0.48 ± 0.05	0.50 ± 0.05
heloc	0.58 ± 0.01	0.67 ± 0.08	0.78 ± 0.09
Bioresponse	0.50 ± 0.02	0.57 ± 0.08	0.59 ± 0.07
electricity	0.54 ± 0.06	0.61 ± 0.08	0.65 ± 0.09
Australian	0.50 ± 0.03	0.54 ± 0.06	0.64 ± 0.08
house_16H	0.38 ± 0.03	0.40 ± 0.07	0.42 ± 0.06
pol	0.17 ± 0.04	0.18 ± 0.05	0.15 ± 0.04
california	0.37 ± 0.03	0.39 ± 0.05	0.40 ± 0.06

Impact of dataset dependent factors Figure 2 highlights how MSBoost and the baseline models perform when noise, number of samples and others are progressively increased on Scikit-Learn's [38] make classification dataset [39]. This is a *cherry-picked* example, but similar trend was found on all other Scikit-Learn's synthetic datasets, their plots can be found in Appendix B.2. Using paired single tail t-test that MSBoost has a p-value < 0.01 when compared to XGBoost and LightGBM for robustness against noise and for impact of number of samples when compared to the baseline models.

[2] The model selection was done on a validation dataset, subsampled from the training data.

[3] For now it wasn't compared to CatBoost, since in order to have a fair comparison, since MSBoost's implementation doesn't have targeted feature encoding for now.

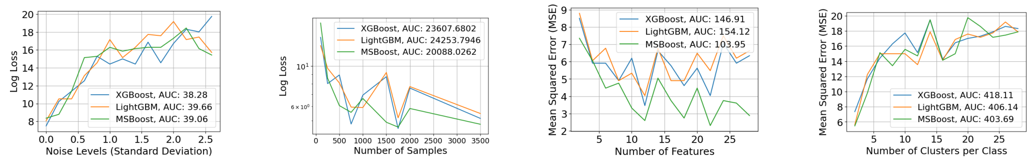


Figure 2: Impact of dataset dependent various factors on log loss for Make Classification Dataset [39]

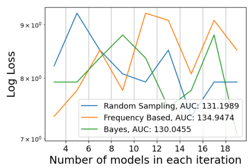


Figure 3: Impact of changing number of models trained, for model selection methods (Scikit-Learn's Make Classification dataset)

Table 3: Comparison on small noisy real world datasets with significant social impact (†MSE & ‡Log Loss)

	UCI ID [40]	**MSBoost**	**LightGBM**	**XGBoost**
AIDS Clinical Trials†	890	8.1e-2 ± 6.5e-3	+7.1% ± +33.8%	+13.5% ± -1.5%
Student Performance†	320	6.3 ± 0.81	+5.4% ± +15.3%	+25.9% ± +42.2%
Energy Efficiency†	242	1.7 ± 0.18	+2.2% ± +16.9%	+34.3% ± +19.1%
Diabetes†	[38]	3017 ± 333	+18.9% ± +30.0%	+32.9% ± +95.4%
Liver Disorders†	60	9.6 ± 1.19	+4.7% ± +2.6%	+23.1% ± -0.3%
Heart Failure Clinical Records†	519	0.12 ± 0.03	+3.4% ± +35.95%	+17.9% ± +37.1%
Thyroid Cancer Recurrence‡	915	1.2 ± 0.88	+30.6% ± -13.1%	+22.9% ± +4.1%
Rice (Cammeo and Osmancik)‡	545	2.79 ± 0.10	+7.7% ± +130.3%	+6.1% ± +31.0%
Blood Transfusion Service‡	176	8.3 ± 0.48	+10.4% ± +0.3%	+13.9% ± +148.0%
Acute Inflammations‡	184	0.0 ± 0.0	0.3 ± 0.6	0.75 ± 1.2
SPECTF Heart‡	96	6.4 ± 1.24	+4.2% ± +93.0%	+2.1% ± +34.9%
Glioma Grading Clinical & ...‡	759	4.8 ± 0.75	+33.9% ± +20.5%	+34.8% ± -26.7%

Impact of model selection methods The effect of number of base models trained on the model selection methods is demonstrated in Figure 3, this is a *cherry-picked* example the rest can be found in Appendix B.3. There is no statistically significant difference in choosing the bayes method over the frequency based method (p = 0.28), but the bayes method turns out to be better than random sampling (p = 0.06).

Social Impact As mentioned above, there is statistically significant evidence that using model selection along with gradient boosting, MSBoost, may improve bias-variance trade-off. Particularly on small and noisy datasets, where usually other machine learning algorithms tend to overfit [41, 42]. Table 3 demonstrates a few possible tabular regression and classification problems with significant social impact, where MSBoost turns out to be better than other methods in terms of MSE/log loss and standard deviation (5 Fold CV).

Limitations (i) Since it trains multiple models for each iteration, MSBoost, has a enormously high time complexity. Where the limiting factor is SVM's RBF kernel, which is quadratic. So the worst case time complexity of MSBoost is approximately $O(n^2)$, whereas LightGBM and others have a time complexity of $O(n\ log\ n)$ (ii) Due to system resource constrains (AMD Ryzen 5 3550H & 8 GB RAM, Ubuntu 22.04.4 LTS), and the enormous time complexity the test most of the benchmarking couldn't be done for more than 1K samples, although this was compensated by benchmarking on 50+ datasets with 5 fold CV.

4 Conclusion

This paper introduces a novel gradient boosting method, MSBoost, which uses model selection to find base estimators for each iteration of gradient boosting. Empirical results show that there is a statistically significant evidence that this method outperforms other popular gradient boosting methods (LightGBM & XGBoost), both in terms of errors and standard deviation of the error. Furthermore, ablation studies reveal that MSBoost outperforms other methods on (synthetic & real) small and noisy datasets, a domain where machine learning algorithms usually struggle. Future work, shall incorporate techiques like targeted feature encoding, GOSS, EFB and other from the current Gradient Boosting methods.

Acknowledgments and Disclosure of Funding

In no particular order, Dr. Arpan Mukherjee (Research Scientist, University at Buffalo) and Rudransh Agnihotri (Founder and CEO of FuturixAI and Quantum Works) are gratefully acknowledged for their suggestions and reviewing this research. Deepika Pareek (Head of Department, Computer Science at Birla Vidya Niketan) is thanked with gratitude for her guidance and encouragement for an initial starting point of this research, which was presented at Regeneron International Science and Engineering Fair (2023). Also parts of the math notation and the table style was adapted from CatBoost [20]. This is a bootstrapped research, and didn't receive any external funding. The author has no competing financial interests.

References

1. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics,* 1189–1232 (2001).
2. Friedman, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis* **38,** 367–378 (2002).
3. Schapire, R. E. The strength of weak learnability. *Machine Learning* **5,** 197–227 (1990).
4. Chen, T. & He, T. *Higgs boson discovery with boosted trees* in *NIPS 2014 workshop on high-energy physics and machine learning* (2015), 69–80.
5. Wu, Z. *et al.* MoleculeNet: a benchmark for molecular machine learning. *Chemical science* **9,** 513–530 (2018).
6. Nobre, J. & Neves, R. F. Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets. *Expert Systems with Applications* **125,** 181–194 (2019).
7. Hajek, P., Abedin, M. Z. & Sivarajah, U. Fraud detection in mobile payment systems using an XGBoost-based framework. *Information Systems Frontiers* **25,** 1985–2003 (2023).
8. Burges, C. J. From ranknet to lambdarank to lambdamart: An overview. *Learning* **11,** 81 (2010).
9. Li, P., Wu, Q. & Burges, C. *McRank: Learning to Rank Using Multiple Classification and Gradient Boosting* in *Advances in Neural Information Processing Systems* (eds Platt, J., Koller, D., Singer, Y. & Roweis, S.) **20** (Curran Associates, Inc., 2007). https://proceedings.neurips.cc/paper_files/paper/2007/file/b86e8d03fe992d1b0e19656875ee557c-Paper.pdf.
10. Gulin, A., Kuralenok, I. & Pavlov, D. *Winning The Transfer Learning Track of Yahoo!'s Learning To Rank Challenge with YetiRank* in *Proceedings of the Learning to Rank Challenge* (eds Chapelle, O., Chang, Y. & Liu, T.-Y.) **14** (PMLR, Haifa, Israel, July 2011), 63–76. https://proceedings.mlr.press/v14/gulin11a.html.
11. Sikander, R., Ghulam, A. & Ali, F. XGB-DrugPred: computational prediction of druggable proteins using eXtreme gradient boosting and optimized features set. *Scientific reports* **12,** 5505 (2022).
12. Sun, X., Liu, M. & Sima, Z. A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters* **32,** 101084. ISSN: 1544-6123. https://doi.org/10.1016/j.frl.2018.12.032 (2020).
13. Natekin, A. & Knoll, A. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics* **7,** 21 (2013).
14. Roe, B. P. *et al.* Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **543,** 577–584 (2005).
15. Wu, Q., Burges, C. J., Svore, K. M. & Gao, J. Adapting boosting for information retrieval measures. *Information Retrieval* **13,** 254–270 (2010).
16. Zhang, Y. & Haghani, A. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies* **58,** 308–324 (2015).
17. Kearns, M. & Valiant, L. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)* **41,** 67–95 (1994).
18. Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system* in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), 785–794.

19. Ke, G. *et al.* Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **30** (2017).
20. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* **31** (2018).
21. Tannor, P. & Rokach, L. *AugBoost: Gradient Boosting Enhanced with Step-Wise Feature Augmentation* in *Twenty-Eighth International Joint Conference on Artificial Intelligence* (July 2019), 3555–3561. `https://doi.org/10.24963/ijcai.2019/493`.
22. Li, B., Friedman, J., Olshen, R. & Stone, C. Classification and regression trees (CART). *Biometrics* **40,** 358–361 (1984).
23. Friedman, J., Hastie, T. & Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* **28,** 337–407 (2000).
24. Rokach, L. & Maimon, O. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **35,** 476–487 (2005).
25. Zięba, M., Tomczak, J. M., Lubicz, M. & Świątek, J. Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. *Applied soft computing* **14,** 99–108 (2014).
26. Caruana, R. & Niculescu-Mizil, A. *An empirical comparison of supervised learning algorithms* in *Proceedings of the 23rd international conference on Machine learning* (2006), 161–168.
27. McElfresh, D. *et al.* When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems* **36** (2024).
28. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* **1,** 67–82 (1997).
29. Wolpert, D. H. The supervised learning no-free-lunch theorems. *Soft computing and industry: Recent applications,* 25–42 (2002).
30. Cortes, C. & Vapnik, V. Support-vector networks. *Machine learning* **20,** 273–297 (1995).
31. Zhou, Z.-H. *Ensemble methods: foundations and algorithms* (CRC press, 2012).
32. Wasserman, L. Bayesian model selection and model averaging. *Journal of mathematical psychology* **44,** 92–107 (2000).
33. Ando, T. Predictive Bayesian Model Selection. *American Journal of Mathematical and Management Sciences* **31,** 13–38. `https://doi.org/10.1080/01966324.2011.10737798` (2011).
34. Chipman, H. *et al.* The practical implementation of Bayesian model selection. *Lecture Notes-Monograph Series,* 65–134 (2001).
35. Vanschoren, J., Van Rijn, J. N., Bischl, B. & Torgo, L. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter* **15,** 49–60 (2014).
36. Bischl, B. *et al.* OpenML: A benchmarking layer on top of OpenML to quickly create, download, and share systematic benchmarks. *NeurIPS.* `https://openreview.net/forum?id=OCrD8ycKjG` (2021).
37. Briscoe, E. & Feldman, J. Conceptual complexity and the bias/variance tradeoff. *Cognition* **118,** 2–16 (2011).
38. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).
39. Guyon, I. *Design of experiments of the NIPS 2003 variable selection benchmark* in *NIPS 2003 workshop on feature extraction and feature selection* **253** (2003), 40.
40. Kelly, M., Longjohn, R. & Nottingham, K. *The UCI Machine Learning Repository* `https://archive.ics.uci.edu`.
41. Lever, J., Krzywinski, M. & Altman, N. Points of significance: model selection and overfitting. *Nature methods* **13,** 703–705. `https://doi.org/10.1038/nmeth.3968` (2016).
42. Oates, T. & Jensen, D. *The Effects of Training Set Size on Decision Tree Complexity* in *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics* (eds Madigan, D. & Smyth, P.) **R1.** Reissued by PMLR on 30 March 2021. (PMLR, Apr. 1997), 379–390. `https://proceedings.mlr.press/r1/oates97b.html`.

43. Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction* https://doi.org/10.1007/978-0-387-84858-7 (Springer, 2009).
44. Hastie, T., Rosset, S., Zhu, J. & Zou, H. Multi-class adaboost. *Statistics and its Interface* **2,** 349–360 (2009).
45. Celeux, G., El Anbari, M., Marin, J.-M. & Robert, C. P. Regularization in regression: comparing Bayesian and frequentist methods in a poorly informative situation (2012).
46. Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics* **19,** 1–67 (1991).
47. Breiman, L. Bagging predictors. *Machine learning* **24,** 123–140 (1996).
48. Marsland, S. *Machine learning: an algorithmic perspective* (Chapman and Hall/CRC, 2011).

Supplementary Materials for MSBoost: Using Model Selection with Multiple Base Estimators for Gradient Boosting

Agnij Moitra
Amity International School (Saket)
agnijmoitra[at]outlook[dot]com

A Pseudo-Code

Algorithm 1 MSBoost Algorithm Pseudocode

Require: Tabular dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, differentiable loss function $\mathcal{L}(\mathbf{y}, F(\mathbf{x}))$, models $\mathcal{M}$, number of iterations N, learning rate α

Initialize $F_0(\mathbf{x}) = \arg\min_k \sum_{i=1}^{n} \mathcal{L}(y_i, k)$
for $i = 1$ to N **do**
    $\mathbf{r}_i = -\left[\frac{\partial \mathcal{L}(\mathbf{y}, F_{i-1}(\mathbf{x}))}{\partial F_{i-1}(\mathbf{x})}\right]$ `// Compute pseudo residuals`
    $h_i(\mathbf{x}) = \arg\min_{\forall M \in \mathcal{M}} \mathcal{L}(\mathbf{y}, M(\mathbf{r}_i))(\mathbf{r}_i)$ `// Choose base estimator for i`$^{\text{th}}$ `layer i.e Model Selection`
    $F_i(\mathbf{x}) = F_{i-1}(\mathbf{x}) + \alpha \cdot h_i(\mathbf{x})$ `// Update model for i`$^{\text{th}}$ `layer`
end for
return $\hat{\mathbf{y}} = F_0(\mathbf{x}) + \sum_{i=1}^{N} F_i(\mathbf{x})$

Algorithm 2 Update Posterior Probabilities for Models

Require: New observed error values $\mathbf{E}$, Prior probabilities for all models $\mathbf{P}$ (For the first iteration it is assumed that all models have an equal prior probability.), Indices of trained models T, Dirichlet prior parameters α, Penalty factor $\beta = 0.7$

$\mathbf{P_T} \leftarrow [P_i \mid i \in \mathbf{T}]$
$\mathbf{S} \sim \mathrm{Dir}(\boldsymbol{\alpha})^{1000}$
$\mathbf{W} \leftarrow []$ `// Initialize weights`
for s $\in$ **S do**
    $w \leftarrow \exp\left(-\sum_{i=1}^{n} \log(s_i) \cdot E_i\right)$ `// Get probabilities`
    $\mathbf{W} \leftarrow \mathbf{W} \cup \{w\}$
end for
$\mathbf{W} \leftarrow \frac{\mathbf{W}}{\sum \mathbf{W}}$
$\mathbf{P'_T} \leftarrow \mathbf{P_T} \cdot (\mathbf{W} \cdot \mathbf{S})$
$\mathbf{P}' \leftarrow \mathbf{P}$ `// Initialize updated posterior probabilities`
for $i \in \mathbf{T}$ **do**
    $P'_i \leftarrow \mathbf{P'_T}[i]$ `// Update posterior probabilities for trained models`
end for
$\mathbf{U} \leftarrow \{i \mid i \notin \mathbf{T}\}$
for $i \in \mathbf{U}$ **do**
    $P'_i \leftarrow P'_i \cdot \beta$ `// Penalize untrained models`
end for
return $\mathbf{P}' \leftarrow \frac{\mathbf{P}'}{\sum \mathbf{P}'}$

B Results

B.1 Benchmarking Results

Table 4: Comparison (regression) with baselines based on mean squared error (MSE)

	MSBoost	LightGBM	XGBoost
wave_energy	0.0 ± 0.0	1.979e+9 ± 2.967e+8	3.007e+9 ± 4.515e+8
SGEMM_GPU_kernel_performance	0.0006 ± 0.0001	0.0021 ± 0.0005	0.0005 ± 0.0001
Friedman 2	150.5209 ± 31.6582	385.8898 ± 57.2398	501.6854 ± 58.3089
Sparse Uncorrelated	1.0338 ± 0.1538	1.5732 ± 0.1136	1.7742 ± 0.2214
kin8nm	0.0217 ± 0.0016	0.0318 ± 0.0015	0.0364 ± 0.0013
sarcos	32 ± 8	46 ± 15	48 ± 10
Moneyball	431.9247 ± 24.2184	588.8988 ± 42.4500	635.3445 ± 39.3275
Parkinsons Telemonitoring	13.5233 ± 2.1851	18.2569 ± 3.4753	13.1871 ± 1.8204
yprop_4_1	0.0007 ± 0.0001	0.0009 ± 0.0001	0.0011 ± 0.0001
fps_benchmark	2354.4510 ± 110.9576	2917.2280 ± 104.0124	3758.8541 ± 395.4950
Zurich Transport	10.0047 ± 0.7051	12.3082 ± 0.9710	15.2101 ± 1.4006
Diabetes	3017.3830 ± 333.9345	3590.3865 ± 433.2183	3991.1318 ± 651.7501
medical_charges	0.0057 ± 0.0021	0.0067 ± 0.0019	0.0068 ± 0.0021
Airlines_DepDelay_1M	3.7258 ± 0.2437	4.3522 ± 0.3432	5.2531 ± 0.3028
visualizing_soil	24.9784 ± 6.9898	28.2629 ± 7.3992	22.4963 ± 9.5333
video_transcoding	146.3238 ± 51.3910	163.5462 ± 60.2620	206.6265 ± 64.0276
health_insurance	310.6828 ± 27.6937	345.2994 ± 31.4665	397.2537 ± 25.3154
grid_stability	0.0009 ± 0.0001	0.0010 ± 0.0001	0.0012 ± 0.0001
abalone	5.3366 ± 0.8634	5.8872 ± 1.2488	6.1480 ± 1.1375
Liver Disorders	9.1786 ± 1.1586	10.0795 ± 1.2242	11.8547 ± 1.1893
student_performance_por	8.1418 ± 1.3556	8.9132 ± 1.2743	12.5153 ± 1.6150
diamonds	1.93e+6 ± 3.44e+5	2.09e+6 ± 4.88e+5	2.30e+6 ± 4.59e+5
auction_verification	9.34e+7 ± 1.15e+7	1.00e+8 ± 9.79e+6	1.52e+8 ± 2.04e+7
cpu_act	10.2950 ± 1.5068	10.7118 ± 1.9016	16.6299 ± 10.0559
Student Performance	6.3849 ± 0.9935	6.5426 ± 0.9590	7.3635 ± 1.2074
pol	103.6665 ± 21.9906	105.9581 ± 42.7356	126.1512 ± 46.4249
AIDS Clinical Trials Group Study	0.0857 ± 0.0065	0.0868 ± 0.0087	13.1871 ± 1.8204
Bike_Sharing_Demand	12307.1564 ± 1427.9618	12461.6368 ± 1476.1868	13478.9580 ± 1027.8582
srsd-feynman_hard	2.549e-70	2.578e-70	2.984e-70
seattlecrime6	151041.8081 ± 3619.6563	151809.1711 ± 3291.9561	151684.2628 ± 3180.8946

Table 5: Comparison (classification) with baselines based on log loss

	MSBoost	LightGBM	XGBoost
phoneme	0.3467 ± 0.0371	0.4324 ± 0.0750	0.4393 ± 0.0623
guillermo	0.5644 ± 0.0461	0.6988 ± 0.1070	0.7725 ± 0.1146
MagicTelescope	0.4020 ± 0.0428	0.4817 ± 0.0510	0.5090 ± 0.0512
heloc	0.5888 ± 0.0130	0.6773 ± 0.0831	0.7849 ± 0.0932
Bioresponse	0.5012 ± 0.0264	0.5705 ± 0.0811	0.5921 ± 0.0752
electricity	0.5462 ± 0.0645	0.6137 ± 0.0818	0.6530 ± 0.0997
Australian	0.5087 ± 0.0318	0.5459 ± 0.0691	0.6432 ± 0.0853
house_16H	0.3847 ± 0.0361	0.4086 ± 0.0702	0.4254 ± 0.0681
pol	0.1738 ± 0.0489	0.1839 ± 0.0598	0.1524 ± 0.0468
Bioresponse	0.5431 ± 0.0805	0.5705 ± 0.0811	0.5921 ± 0.0752
california	0.3736 ± 0.0345	0.3911 ± 0.0552	0.4050 ± 0.0625
heloc	0.6507 ± 0.0872	0.6773 ± 0.0831	0.7849 ± 0.0932
higgs	0.7332 ± 0.1404	0.7543 ± 0.0862	0.8221 ± 0.1008
compas-two-years	0.6986 ± 0.1059	0.7138 ± 0.0543	0.8056 ± 0.0742
Higgs	0.7255 ± 0.0842	0.7409 ± 0.0540	0.8661 ± 0.0510
MiniBooNE	0.2995 ± 0.0294	0.3043 ± 0.0568	0.3087 ± 0.0610

Table 6: (Absolute values) Comparison on small noisy real world datasets with significant social impact (†MSE & ‡Log Loss)

	UCI ID [40]	MSBoost	LightGBM	XGBoost
AIDS Clinical Trials†	890	8.1e-2 ± 6.5e-3	8.7e-2 ± 1.1e-2	9.2e-2 ± 8.6e-3
Student Performance†	320	6.3 ± 0.81	6.6 ± 0.93	7.9 ± 1.15
Energy Efficiency†	242	1.7 ± 0.18	1.74 ± 0.21	2.28 ± 0.26
Diabetes†	[38]	3017 ± 333	3585 ± 433	4010 ± 651
Liver Disorders†	60	9.6 ± 1.19	10.05 ± 1.22	11.82 ± 1.18
Heart Failure Clinical Records†	519	0.12 ± 0.03	0.124 ± 0.041	0.141 ± 0.047
Thyroid Cancer Recurrence‡	915	1.2 ± 0.88	1.57 ± 0.77	1.47 ± 0.92
Rice (Cammeo and Osmancik)‡	545	2.79 ± 0.10	3.00 ± 0.23	2.96 ± 0.13
Blood Transfusion Service‡	176	8.3 ± 0.48	9.16 ± 0.48	9.46 ± 1.19
Acute Inflammations‡	184	0.0 ± 0.0	0.003 ± 0.006	0.0075 ± 0.012
SPECTF Heart‡	96	6.4 ± 1.24	6.67 ± 1.2	6.53 ± 1.29
Glioma Grading Clinical & ...‡	759	4.8 ± 0.75	6.43 ± 0.9	6.47 ± 0.55

B.2 Impact of Data Dependent Factors

This section contains all the plots[4] for impact of data dependent factors on Scikit-Learn's [38] simulated datasets. Lower area under the loss curve indicate better performance.

B.2.1 Classification Datasets[5]

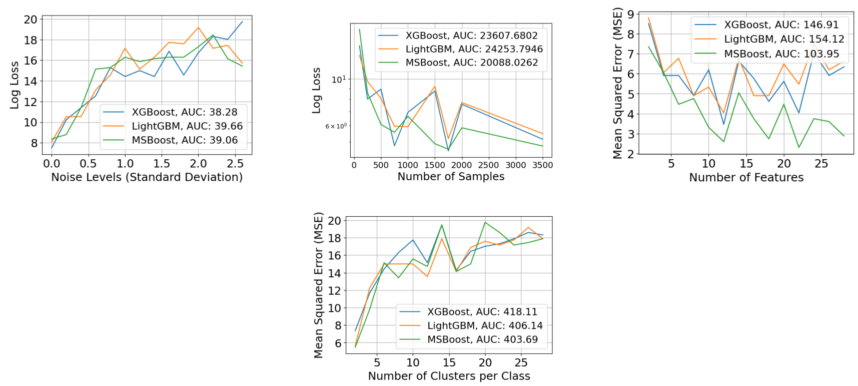


Figure 4: Make Classification [43]

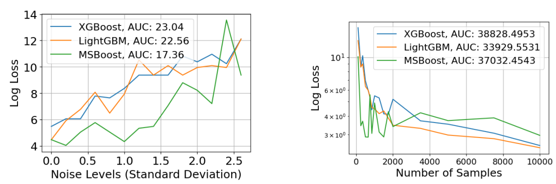


Figure 5: Hastie 10 Dataset [43]

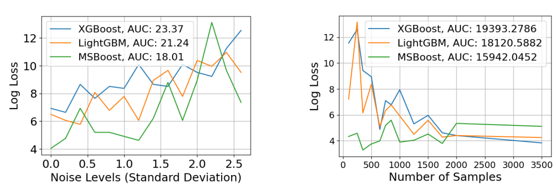


Figure 6: Gaussian Quantiles [44]

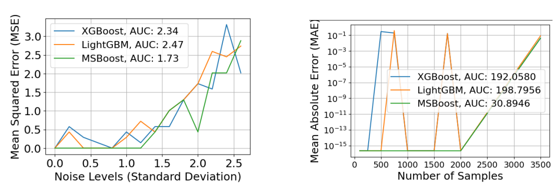


Figure 7: Make Blobs

[4]The exact values for the x and y coordinates can be found in `https://github.com/Agnij-Moitra/MSBoost`

[5]Due to computational and hardware constrains the jupyter kernel crashed when the number of samples went more than around 5K, so it wasn't done on 10K samples like regression.

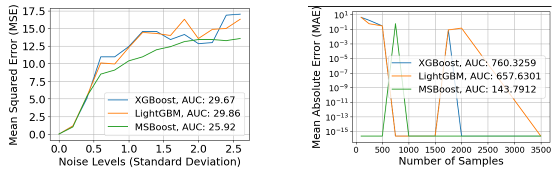


Figure 8: Make Moons

B.2.2 Regression Datasets

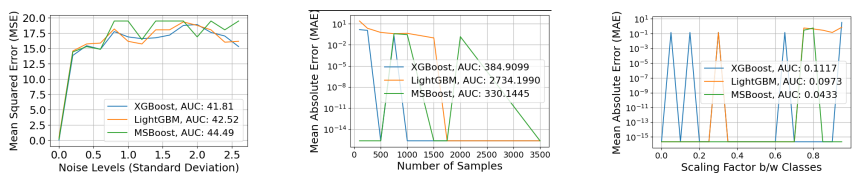


Figure 9: Make Circles

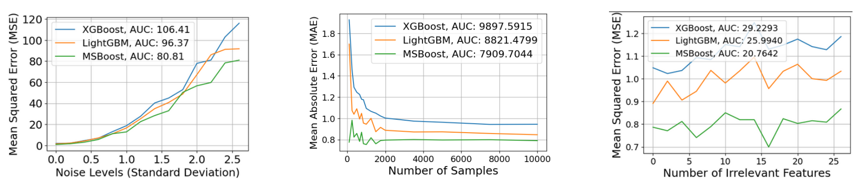


Figure 10: Sparse Uncorrelated [45]

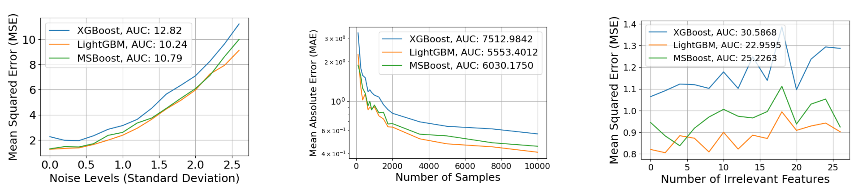


Figure 11: Friedman 1 [46, 47]

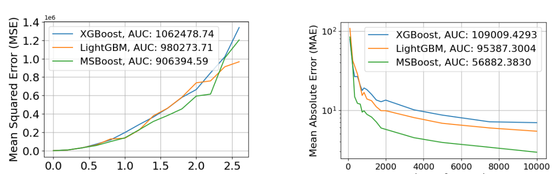


Figure 12: Friedman 2 [46, 47]

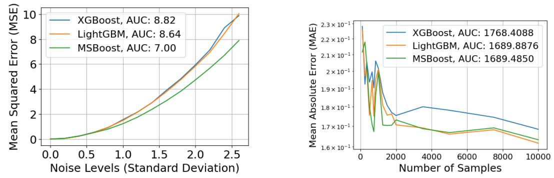


Figure 13: Friedman 3 [46, 47]

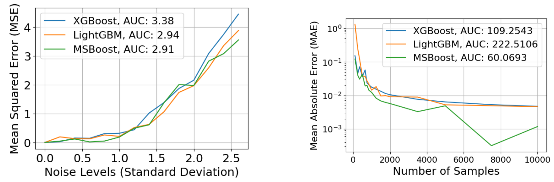


Figure 14: Swiss Roll [48]

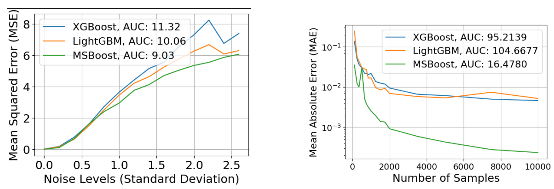
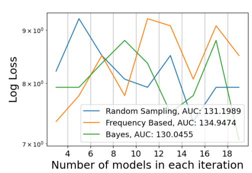
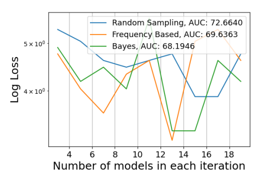


Figure 15: S Curve

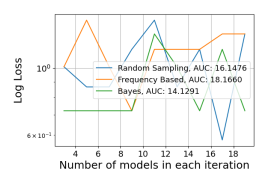
B.3 Impact of Model Selection Methods



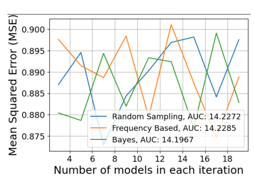
(a) Make Classification



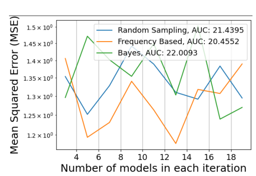
(b) Make Hastie



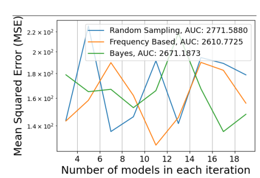
(c) Make Gaussian Quantiles



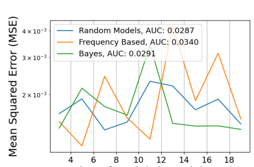
(d) Sparse Uncorrelated



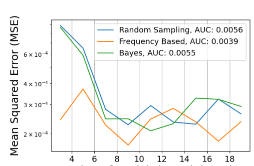
(e) Friedman 1



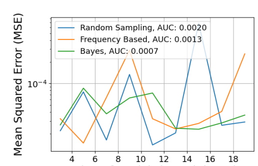
(f) Friedman 2



(g) Friedman 3



(h) Swiss Roll



(i) S Curve

Figure 16: Impact of number of models for model selection methods

Table 7: p-values for impact of number of models on model selection methods (Row vs. Column)

	Bayes	Frequency Based	Random Sampling
Bayes	1.00	0.71	0.82
Frequency Based	**0.28**	1.00	0.82
Random Sampling	**0.06**	0.17	1.00

C Dataset Sources & Description[6]

C.1 Benchmarking Datasets

This section contains descriptions for selected datasets used for benchmarking. Please refer to the original sources [35, 36, 38, 40] for descriptions for all the datasets used in Table 4 & 5.

- wave_energy: This data set consists of positions and absorbed power outputs of wave energy converters (WECs) in four real wave scenarios from the southern coast of Australia. The data is obtained from an optimization method (blackbox optimization) with the goal of finding the optimal buoys placement. Each instance represents wave energy returns for different placements of 16 buoys.
- Friedman 2: $y(X) = (X[:,0]^2 + (X[:,1] \times X[:,2] - \sqrt{\frac{1}{X[:,1]*X[:,3]))^2}}$ + noise $\times\, N(0,1)$
- Sparse Uncorrelated: $X \sim N(0,1); y(X) = X[:,0] + 2 \times X[:,1] - 2 \times X[:,2] - 1.5 \times X[:,3]$
- kin8nm: A realistic simulation of the forward dynamics of an 8 link all-revolute robot arm. The task is to predict the distance of the end-effector from a target based on angular positions of the joints.
- sarcos: Dataset related to an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm. Predict joint torques from joint positions, velocities, and accelerations.
- Moneyball: Dataset used in baseball analytics, focusing on statistics like on-base percentage (OBP) and slugging percentage (SLG) to predict player performance.
- yprop_4_1: Dataset used in the tabular data benchmark, transformed accordingly, for regression on categorical and numerical features.
- fps_benchmark: Dataset containing FPS measurements of video games executed on computers, characterized by CPU and GPU specifications and game settings.
- Zurich Transport: Zurich public transport delay data, cleaned and prepared for analysis.
- phoneme: Dataset to distinguish between nasal (class 0) and oral sounds (class 1) using harmonics and energy ratios.
- guillermo: The challenge introduces diverse, real-world datasets formatted uniformly for binary classification tasks, evaluated by AUC. Participants use preprocessed matrices and adhere to time-constrained evaluations on Codalab.
- MagicTelescope: Simulation data from a ground-based atmospheric Cherenkov gamma telescope, detecting high-energy gamma particles.
- heloc: Dataset used in the tabular data benchmark, transformed accordingly, for classification on numerical features.
- Bioresponse: Predict biological responses of molecules based on chemical properties and molecular descriptors.
- electricity: Dataset collected from the Australian New South Wales Electricity Market, containing 45,312 instances over a period from 7 May 1996 to 5 December 1998.
- Australian: Australian Credit Approval dataset, anonymized and converted to ARFF format, used in credit card application analysis.
- house_16H: Binarized version of the house dataset, converting numeric target features to a two-class nominal target feature based on mean values.
- pol: Dataset used in the tabular data benchmark for classification on numerical features, related to a telecommunication problem.
- california: The dataset includes data from all California block groups in the 1990 Census, averaging 1425.5 individuals per group in compact areas varying with population density. It features 20,640 observations across 9 variables, excluding groups with zero entries, with the dependent variable being ln(median house value).

[6]GPT-3.5 was used to summarize the data description from original sources.

C.2 Social Impact Datasets

- AIDS Clinical Trials Group Study 175: The AIDS Clinical Trials Group Study 175 Dataset contains healthcare statistics and categorical information about patients who have been diagnosed with AIDS. This dataset was initially published in 1996. The prediction task is to predict whether or not each patient died within a certain window of time or not.
- Student Performance: The dataset analyzes student achievement in two Portuguese secondary schools, covering grades, demographics, and school-related factors. It includes separate datasets for Mathematics (mat) and Portuguese language (por), with a strong correlation between final grade (G3) and earlier grades (G2 and G1), essential for prediction and analysis according to Cortez and Silva (2008).
- Energy Efficiency: The dataset consists of 768 samples representing 12 different building shapes simulated in Ecotect. Variations include glazing area, distribution, orientation, and other parameters, generating 8 features per sample. The objective involves predicting two real-valued responses or, alternatively, using the rounded responses for multi-class classification.
- Diabetes: Contains 442 samples with 10 numeric features related to diabetes progression, including age, sex, BMI, blood pressure, and blood serum measurements. Target is a continuous measure of disease progression one year after baseline.
- Liver Disorders: The dataset contains records of male individuals with 5 blood test variables possibly related to liver disorders from alcohol consumption. The 7th field serves as a train/test selector, not as a dependent variable for liver disorder presence/absence; researchers should use the dichotomized 6th field (drinks) for classification.
- Heart Failure Clinical Records: This dataset contains the medical records of 299 patients who had heart failure, collected during their follow-up period, where each patient profile has 13 clinical features.
- Differentiated Thyroid Cancer Recurrence: This data set contains 13 clinicopathologic features aiming to predict recurrence of well differentiated thyroid cancer. The data set was collected in duration of 15 years and each patient was followed for at least 10 years.
- Rice (Cammeo and Osmancik): A study was conducted on Osmancik and Cammeo rice species, prominent in Turkey since 1997 and 2014 respectively. 3810 rice grain images were analyzed, deriving 7 morphological features per grain. Osmancik grains are noted for their wide, long, glassy, and dull appearance, while Cammeo grains exhibit similar characteristics with a focus on width and length.
- Blood Transfusion Service Center: This study utilized data from the Blood Transfusion Service Center in Hsin-Chu City, Taiwan, for a classification problem. The dataset comprises 748 donor records selected randomly, with features including R (Recency), F (Frequency), M (Monetary), T (Time since first donation), and a binary variable indicating blood donation in March 2007 (1 for donated, 0 for not donated). The objective was to develop a RFMTC marketing model using these variables.
- Acute Inflammations: The dataset was crafted by a medical expert to support an expert system for diagnosing two urinary system diseases: acute inflammation of the urinary bladder and acute nephritis. It utilizes Rough Sets Theory for rule detection, with each instance representing a potential patient.
- SPECTF Heart: Data on cardiac Single Proton Emission Computed Tomography (SPECT) images. Each patient classified into two categories: normal and abnormal. The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature pattern was created for each patient.
- Glioma Grading Clinical and Mutation Features: The dataset focuses on gliomas, primary brain tumors graded as LGG (Lower-Grade Glioma) or GBM (Glioblastoma Multiforme), based on histological/imaging criteria and molecular mutations. It includes the most frequently mutated 20 genes and 3 clinical features from TCGA-LGG and TCGA-GBM projects. The goal is to predict the glioma grade (LGG or GBM) using these features, aiming to identify the optimal subset for improved diagnostic accuracy and cost reduction in molecular testing for glioma patients.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the claims stated in the abstract and introduction accurately reflect the paper's contribution and scope, and a separate section discusses the limitations. The benchmarking was done on around 50+ datasets with $p < 0.01$ for all the major claims so the claims are expected to generalize to other settings and real world usage.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations section discusses all the possible limitations that the author is aware of.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper entails an informal intuitive justification, and the supplementary material contains a formal proof by induction to provide theoretical justification for the algorithm.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All information required to reproduce the results have been released in the Github repository, including a random seed to get the precise values from the benchmarking.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: (i) The Github repository contains all the commands and environments required to reproduce the results. (ii) The jupyter notebooks contains the repository contains the data ID(s) which were used to fetch the benchmarking datasets from Scikit-Learn, OpenML, and UCI's Machine Learning Repository, and the automated pre-processing steps taken.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The testing was done using 5 Fold cross validation, and only default hyper parameters were used due to hardware resource constrains. Additional details can be found in the jupyter notebooks, from the Github repository, where the tests were performed.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Paired single tailed t-tests where performed, excluding extreme outliers, wherever it was required to prove the statistically significance of the claims presented, and standard deviations from the 5-fold cross validation have been reported.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Yes, the paper does list the compute resources used, and with the random seed with was set to 7 at the beginning of each experiment all the experiments can be reproduced, via the github repository. For the benchmarking only default hyper-parameters were used. Unfortunately this paper didn't note the precise run times, but the main regression and classification benchmarks had run overnight. And the ablations studies took around 12 hours of wall time.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, this paper does conform to every aspect of the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes both the positive and negative social impacts have been discussed.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Technically a tabular machine learning algorithm could be used for malicious purposes, but it does not pose a significantly high risk like LLM(s), image generators *et cetera*, and like previous methods (XGBoost, LightGBM, CatBoost) the algorithm MSBoost is freely available under an open-sourced licensed. All the tabular datasets used are anyways anonymized and already open-sourced and can be accessed via the Scikit-Learn, UCI and OpenML API(s), so any safegaurds were not required from our end for the datasets.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: A few code snippets from Github and Stackoverflow were used and they have been credited appropriately in the docstrings of the functions/classes where they were used or with in-line comments. The major datasets used have been cited explicitly, but it not feasible to cite all 50+ individual datasets. So UCI's Machine Learning Repository, and OpenML have been cited explicitly as they had requested. Baseline models like LightGBM, XGBoost and CatBoost have been cited, and Scikit-Learn which was used extensively for this has also been cited.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All images and code released under this paper can be used whilst adhering to Apache 2.0 license.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: This paper does not involve crowdsourcing nor research with human subjects.