Model Evaluator Overcomes Flaws in Conventional ML Metrics Comparison

Table of Contents

Abstract

Artificial Intelligence and Machine Learning (AI/ML) are probably the most popular technological buzzwords for the last 2-3 decades. These technologies have experienced the highest growth in the past 10 years. It won't be an overstatement to say that the future of AI/ML seems to be even more exciting.

AI's prediction quality and ML's capability are gradually improving with time. However, because there is no efficient way to eliminate the unnecessary metrics, confirm model failure, and optimize the input training set the current methods that evaluate ML training models are flawed. The way they gauge the accuracy of predictions are limited and often miss to reveal the true performance of an ML model.

The low-quality metrics in the model evaluation may lead to multiple challenges for ML projects, such as overloaded databases, less precise decisions, compromised model performance, and so on. And that's where we need a quality-driven model evaluation solution.

This paper proposes a Model Evaluator (ME) to combat the above-mentioned challenges for AI/ML developers and innovators. To be specific, it covers shortcomings of the current metrics used, tells how the ME package can help overcome them, and presents an example case study to explain the proposed solution better. The paper concludes with a summary of future roadmaps and goals for the package and invites contributions.

Introduction

While the overall state of the art for AI/ML has advanced greatly over the past years, the same cannot be said for making meaningful and statistically-sound comparisons and evaluations of estimators[i].

Among the many reasons that statistical significance and Bayesian comparison or similar robust measures have not been broadly adopted are the uncertainty and caveats around most of the proposed solutions. In many cases, the recommendations are not definitive but couched in guidance that leaves much to be decided by the practitioner.[ii]

Another problem with comprehensively applying statistical significance to ML applications is the limited attention paid to regression. Early papers emphasized too little on the time series regression. In many cases, tests were designed for classification only, such as McNemar's test [6].

Modifying these tests for regression can be problematic and may lead to errors. So, we need a proper method for model evaluation that is designed considering regression as well as classification problems.

Background

One of the great strengths of AI/ML practice, since its earliest incarnations, is the reliance on measurements to identify superior solutions. A multitude of metrics are available for both classification and regression that help

---

[i] Estimator refers to any function that performs fitting and analysis of datasets. Estimators are tailored to certain use cases, such as classification or regression, time series or tabular data, and algorithms, from tree models to deep learning.

[ii] The researcher advised against a particular test because it entailed too much risk of false positive results. However, if the study emphasized the importance of avoiding false negatives, perhaps for disease, the test could be used. While the nuance may be interesting, it adds uncertainty to the decision: Exactly how much emphasis on false negatives is sufficient to justify the test?

experts compare outcomes. For example, professionals often utilize techniques ranging from classification accuracy to mean squared error for regression.

Despite the range of choices, the mainstream approach for comparing ML models is the same. We often designate our options and then choose between them. The process may use different estimators based on a simple comparison of metric values [3]. However, this approach entails several shortcomings.

- **Apparent Superiority due to Selected Test**

Simply choosing the metric with the best score, say in comparing two R-squared scores[19], does not guarantee that the selected model or estimator is truly superior. For example, if the R values for the two models are 0.75 and 0.8, the difference may simply be random due to the selected test set.

Cases like the one above negates the supposed superiority of the higher-scoring model. Another example of such false apparent superiority is shown in the **Visual Summary** section.

- **Flaws Induced by Feature Selection**

The selection risk grows if several features are being evaluated, for example, in the case of hyper-parameter optimization. Simply choosing the set with the best score does not reflect the overall interaction effects between the various features, which could show that a particular feature is more important overall, but, by chance, a sub-optimal value for the feature is chosen by relying on a single metric.

- **Over-reliance on NHST**

Further uncertainty arises due to excessive dependency on the null hypothesis testing (NHST). NHST is sensitive to sample size and cannot determine the practical importance of the statistical relations. It may give false positives in various scenarios, leading to bad decisions[4]. For example, the so-called p-value, which determines if the null hypotheses is rejected, no matter how large it is, in no way distinguishes good evidence for H0 from not much evidence at all.

Importance of a Good Machine Learning Model

Evaluating the model incorrectly and going ahead with a flawed model may result in various issues[18], such as:

- Inability to find the relationship between variables
- Faulty or biased predictions
- Need for excessive training data
- Inability to remove outliers from the training data set
- Insufficient evaluation of training data[iii]
- Slow training time performance due to data overload[iv]

[iii] A early selection of a (non-significant) model hinders further evaluation and discovery of other better options

[iv] As an efficient model may need a lot of data for training, it will have overloaded data, resulting in its poor performance.

On the contrary, a good ML model is very useful for its end-users. It can help users make better decisions and eliminate manual operations from the process. Selecting a good ML model has various benefits, including –

- **Efficient ML Solutions**

The main aim of evaluating and comparing ML models is to ensure that the machine learning solution/product (utilizing this model) is robust and performant. If the best model is selected for use, the ML algorithm will fulfill the business/user requirements more efficiently.

- **Longevity**

A problematic model that fails to interpret the unseen data and has tight coupling cannot be used for too long in practice. It will lead to faulty decisions and incorrect outputs as the data set grows with time.

To predict correctly, a model must have the ability to understand the underlying data patterns, which only an effective model can do.

Additionally, the need for frequent re-training can be eliminated for such a model.

- **Simplified Data Error Detection and Model Re-training**

If there is a fault in the training data being fed to a model, the developer will be able to detect this problem early. This allows the developer to explore ways to improve the data's predictive value through techniques such as feature engineering, hyper-parameter optimization and bootstrapping. At the same time, he can clean the previous data with anomaly detection and other methods for pre- and post-processing, and feed the new data for re-training the model more effectively.

- **High Processing Speed**

Evaluating various models and selecting the best one can improve the processing/operation speed for your ML algorithm/solution. A good model will have sufficient resources, allocated assets, and production ability to perform as per an organization's needs. Also, it will use an optimal amount of training to start producing accurate results – earlier than a less efficient model.

Model Evaluator: A Better Way

Considering the need for a reliable way to assess a model, this paper proposes the **Model Evaluator (ME)** as a solution. ME is a new Python package designed to perform meaningful testing. It uses a mix of tests to identify optimal choices of such elements as hyper-parameter tuning values.

The ME package does so by identifying those metric values that deliver not just a superior metric value but true statistically significant superiority. It is explicitly built to overcome the flaws associated with the current simplistic approach to metric comparison.

Motivated by the need for a comprehensive and statistically sound approach to identifying truly superior choices from among estimators and estimator parameters, the developers of Model Evaluator (ME) identified four primary elements required to meet this objective:

1. Test pre-existing analysis results for true statistical significance (S/S).
2. Apply multiple metric-agnostic tests.
3. Identify statistically significant options, that is, those estimators/parameters that demonstrate statistically confirmed differences.

4. Automate selection of S/S options. Assuming more than one estimator/parameter combination demonstrates statistical significance, provide a user-settable or customizable algorithm to select between the options.

The primary goal of creating ME was to meet these objectives and improve the model evaluation process. This package provides the following to its adopters:

1. A methodology to evaluate S/S among and between estimators and associated parameters
2. A comprehensive set of evaluation functions (*meta*-tests) to assess S/S for all types of estimators and dataset types, both tabular and time series, regardless of the specific metric(s) applied by the researcher.
3. A simple API for users to load analysis results and supporting dataset(s) to ME for evaluation.

Methodology Foundations

To achieve the objectives specified in the previous section, ME needs to employ best practices and tools. Many of the existing tests considered for ME were crafted years, if not decades, ago. As these methods were designed before the advent of current approaches such as Bayesian and Causal analysis and improvements in computing power, they might not be useful for various scenarios today.

A primary focus in Model Evaluator development was to identify methods and tests that demonstrate proven ability to discern statistical significance. After evaluating research from many sources, three stood out: Dietterich [5], Raschka [13], and Varoquaux and Colliot [15]. These researchers and ME share a common goal of improving the quality and utility of ML/AI analysis and inference on a statistically-sound foundation.

Based on these sources, three tests were incorporated into ME (see *Evaluation Methods* below). While these (meta-) tests are proven effective, they generally rely on NHST. Given the limitations of NHST[1][8][9], ME is moving to incorporate Bayesian methods[10][14]. Other promising techniques, such as Causal Inference[23], will follow.

Bayesian analyses have become increasingly popular in recent years due to the ongoing discussion about statistical significance [2]. Inference using significance testing and Bayes factors are compared in several case studies based on real research[7]. The first study illustrates that the methods will often agree, both in motivating researchers to conclude that hypothesis 1 (H1) is supported better than hypothesis 0 (H0) and the other way round, that H0 is better supported than H1. However, there are also cases where the methods will disagree. In such cases, the study indicates Bayesian methods demonstrate two main advantages to NHST: 1) assessing the probability of the alternate hypothesis (H1) and 2) controlling false positives. [9]

Causal inference, on the other hand, models the outcome of interventions and formulize the counter-factual reasoning. It allows drawing causal conclusions.

The rest of this paper will discuss ME in more detail and present examples of its use.

The Modus Operandi of ME

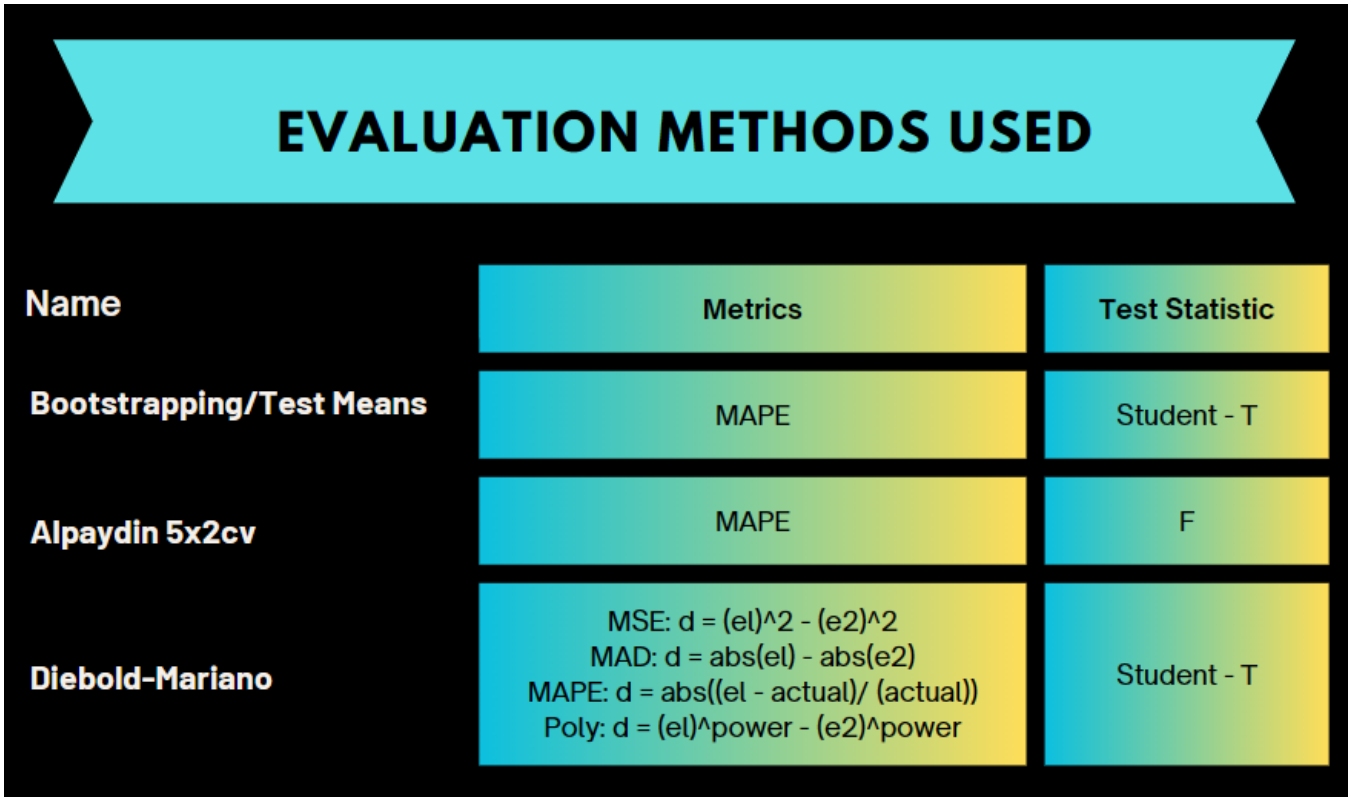
ME Process Overview

The steps that ME typically follows are mentioned below:

- Apply meta-tests to pairs of candidate estimators.[v]
- For each pair, determine if estimators produce statistically significant differences
- Compare overall estimator performance
- Select the best estimator[vi]

Evaluation Methods (*Meta-Tests*)

For ME, the aim of developers is to use any metric-agnostic process or methodology that explicitly assesses statistical significance. Though it currently deploys three meta-tests (Bootstrapping/Test Means[20], Alpaydin 5x2cv, and Diebold-Mariano[21] - see table), there will be more tests in the process. The tests to be added are either in development or in evaluation.[vii]



Dataset Used

We are using the well-known Boston housing dataset from Kaggle.com[22] considering its popularity and reliability. It has 506 observations and 14 variables, alongside a few missing values. It is available in public domain, under the license CC0 (Creative Commons).

---

[v] In actual use, ME users will provide these estimators and associated data set(s) as input.

[vi] IMPORTANT: The estimator is a generic term for a regression or classification model. We can compare different estimators or different versions of the same estimator.

[vii] Current methods are frequentist, but Bayesian will be added in the next release.

Applying the Methods

While methods are important, even more so is their application. How does a user prepare for applying ME to their existing estimators and associated datasets? Let's have a look.

Preparation for Assessing Statistical Significance

- Develop candidate estimators (regression or classification) as needed to create an ML solution.
- Train estimators against a designated data set (tabular or time series) using any metrics[viii]

Apply ME meta-tests to user-selected estimators and data sets

- The user loads estimators and data sets into ME.
- ME creates pairwise combinations[ix] of all estimators for evaluation.
- ME evaluates each pair for significant differences while applying multi-test adjustments.[x]
- ME records and displays results.
- The user selects the best estimators for further analysis and/or deployment.

Example: Regression on Tabular Data

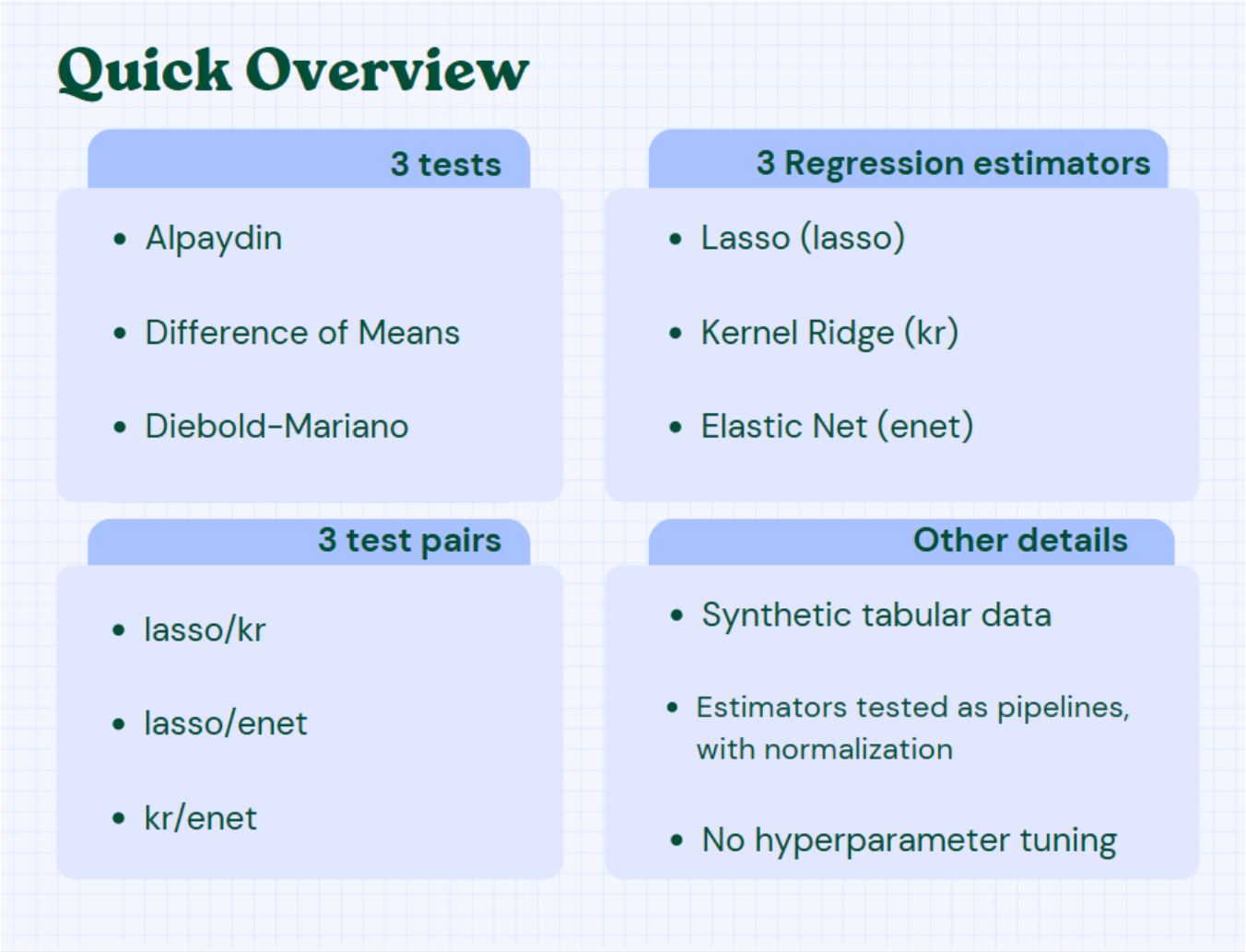
This experiment utilized 3 tests (Alpaydin, Diff. of Means, and Diebold-Mariano) while using three regression estimators (lasso, kr, and enet) for the synthetic tabular data. The 3 test pairs, created by combining diverse regression estimators (lasso/kr, lasso/enet, and kr/enet), were estimated in the process.

Estimators were tested as pipelines (with normalization). No hyperparameter tuning was performed during the tests.

[viii] In actual use, these steps would be performed by the user, such as a data analyst or statistician. ME includes built-in data sets and estimators to demonstrate its function.

[ix] Pairwise evaluation has several benefits, including demonstrating their relative performance as measured by different methods.

[x] ME performs multiple tests for significance. Given the opportunity for false positive results in such circumstances, the best practice is to apply more stringent standards, such as Bonferroni correction or family error rate.



The researcher calculated six total metrics for each comparison. Here, one method, Diebold-Mariano, applies four metrics. Alpaydin and the difference of means methods produced one metric each.

Complete Results Summary

This section comprises several tables displaying the result fetched for the above example. Please note that cells with _sig suffix display significance status.

Result: Alpaydin F-Test

ALPAYDIN F-TEST

ESTIMATOR PAIRS	f_test-stat	f_test-p-val	f-test_confidence	f_test-sig	f_test-sup_est
Elastic_Net - Kernel_Ridge	1.58	0.3204	0.95	FALSE	Elastic_Net
Elastic_Net - LASSO	0.69	0.7111	0.95	FALSE	Kernel_Ridge
Kernel_Ridge - LASSO	66.61	0.0001	0.95	TRUE	Elastic_Net

Result: Difference of Means Test

DIFFERENCE OF MEANS TEST

ESTIMATOR PAIRS	dif_mns_t_stat	dif_mns-p-val	dif_mns_confidence	dif_mns-sig	dif_mns-sup_est
Elastic_Net - Kernel_Ridge	10.65	1.59754E-24	0.95	TRUE	Elastic_Net
Elastic_Net - LASSO	7.22	2.18148E-12	0.95	TRUE	LASSO
Kernel_Ridge - LASSO	-8.68	1.25496E-11	0.95	TRUE	Elastic_Net

Result: Diebold-Mariano Test

DIEBOLD-MARIANO TEST

ESTIMATOR PAIRS	dm_MAD-stat_val	dm_MAD-p-val	dm_MAD-confidence	dm_MAD-sig	dm_MAD-sup_est
Elastic_Net - Kernel_Ridge	5.95	0.0000	0.95	TRUE	Elastic_Net
Elastic_Net - LASSO	-36.66	0.0000	0.95	TRUE	Kernel_Ridge
Kernel_Ridge - LASSO	-68.93	0.0000	0.95	TRUE	Elastic_Net

Criterion/Metric: Mean Absolute Deviation

ESTIMATOR PAIRS	dm_MAPE-stat_val	dm_MAPE-p-val	dm_MAPE-confidence	dm_MAPE-sig	dm_MAPE-sup_est
Elastic_Net - Kernel_Ridge	-10.27	0.0000	0.95	TRUE	Kernel_Ridge
Elastic_Net - LASSO	-19.69	0.0000	0.95	TRUE	Kernel_Ridge
Kernel_Ridge - LASSO	-25.46	0.0000	0.95	TRUE	Elastic_Net

Criterion/Metric: Mean Absolute Percentage Error Value

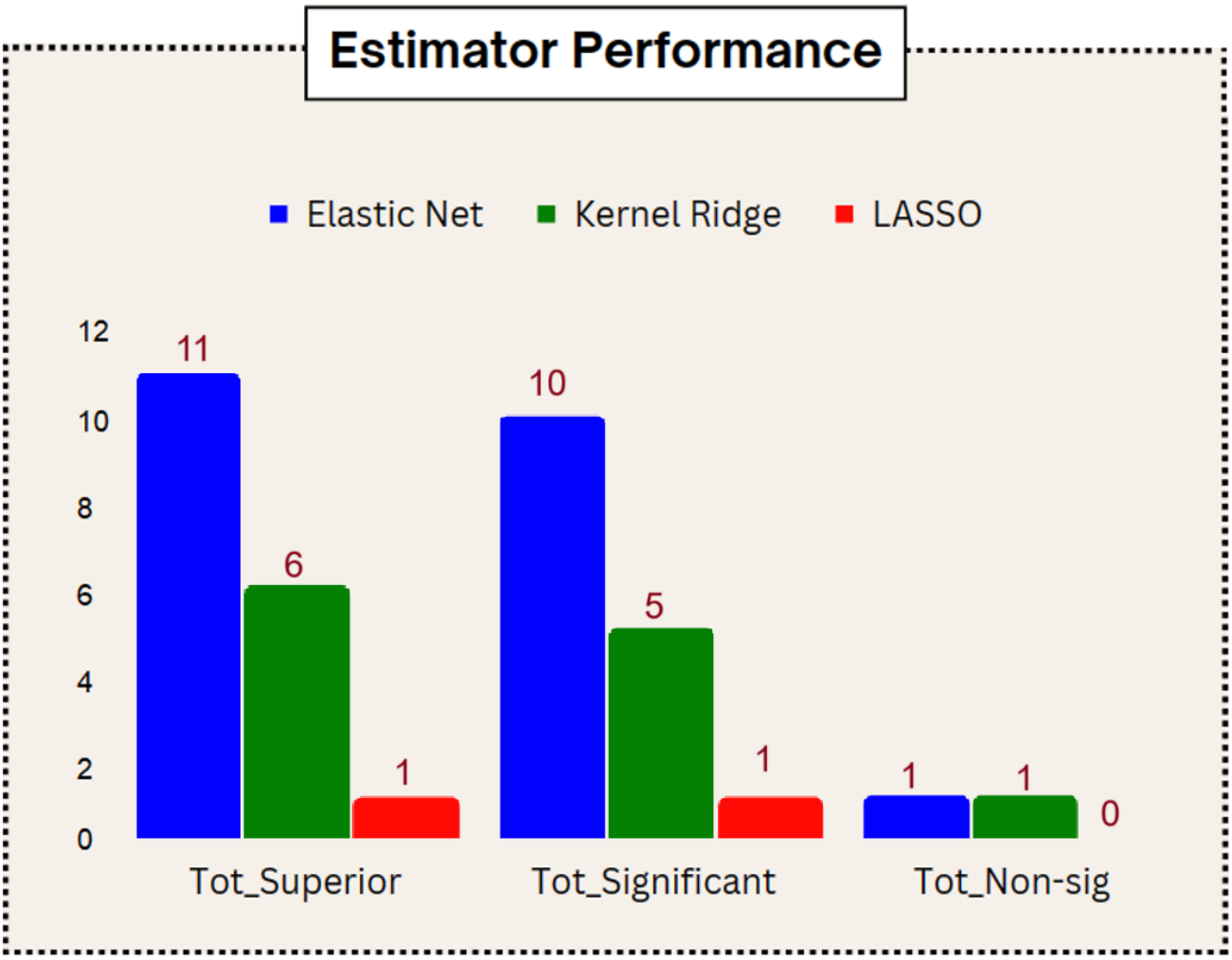
ESTIMATOR PAIRS	dm_MSE-stat_val	dm_MSE-p-val	dm_MSE-confidence	dm_MSE-sig	dm_MSE-sup_est
Elastic_Net - Kernel_Ridge	3.85	0.0001	0.95	TRUE	Elastic_Net
Elastic_Net - LASSO	-21.83	0.0000	0.95	TRUE	Kernel_Ridge
Kernel_Ridge - LASSO	-28.28	0.0000	0.95	TRUE	Elastic_Net

Criterion/Metric: Mean Squared Error

ESTIMATOR PAIRS	dm_Poly-stat_val	dm_Poly-p-val	dm_Poly-confidence	dm_Poly-sig	dm_Poly-sup_est
Elastic_Net - Kernel_Ridge	16.88	0.0001	0.95	TRUE	Elastic_Net
Elastic_Net - LASSO	15.98	0.0000	0.95	TRUE	Kernel_Ridge
Kernel_Ridge - LASSO	14.49	0.0000	0.95	TRUE	Elastic_Net

Criterion/Metric: Poly Statistic Value

Visual Summary



This chart shows comparative estimator results in three categories.

1. **All:** It is the count of estimators' metric-based pairwise performance.
2. **SIG:** SIG stands for Statistically superior (S/S) results. Not all superior values are significant.
3. **Non-SIG:** It specifies the counts of non-s/s results.

Result Interpretation

The most important interpretation from these results is that no estimator is definitely superior. Each presents superior outcomes for some tests, but not all. Can (or should) the researcher select one of these estimators for her application?

Discussion

Considering the previous section, we need to keep in mind that such an outcome adds uncertainty in selecting the best estimator. It also reminds us that finding a signal in the noise is challenging and is certainly to be desired as opposed to not having such information.

Now, to do well in this scenario, a few suggestions are enlisted below:

1. Introduce new data sets and/or estimators to expand results and see if a clear winner emerges
2. Select an estimator based on –
    a. The highest number of total wins (All)
    b. The highest number of statistically significant wins (SIG)
3. Calculate the number of wins/losses by pair and compare to determine if an estimator shows consistent superiority over the others. Use the summary results table (above), which is auto-generated for each analysis.
4. Create a *Desirability Index* [16] to aggregate performance into a single comparative metric for each estimator. This is particularly useful when comparing many estimators.

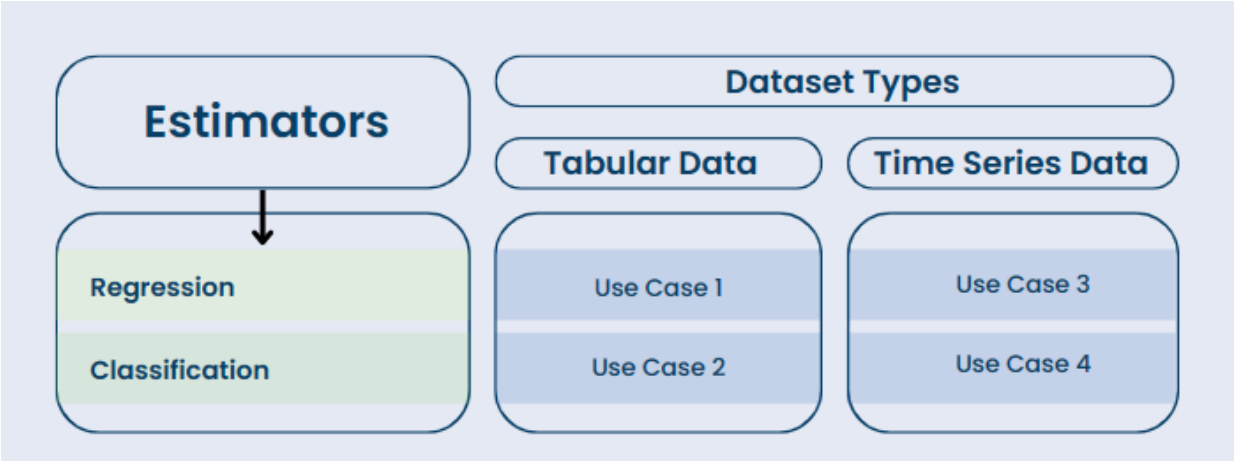
Going forward, ME will automate the calculation of some or all of the above.

Technical Challenges

To be fully effective, the proposed Model Evaluator (ME) must support four use cases: combining estimators and dataset types:

1. **Estimators:** Regression and Classification
2. **Dataset Types:** Tabular and Time Series

See this image to understand the 4 use cases better.



While researchers are free to select the appropriate combination for a study, ME evaluation methods support all four combinations of estimators and dataset types when assessing statistical significance. ME selects the appropriate combination for the researcher's study.

The original methods did not support time series explicitly. Doing so requires significant changes to the entire analysis process – all focused on ensuring that the temporal dimension of the time series is honored and informs the analysis and post-processing inference.

An in-depth discussion of time series protocols is beyond the scope of this paper, but here are some of the associated challenges and changes:

1. Incorporating the differences between tabular predictions and time series forecasts
2. No shuffling of samples
3. Time-sensitive versions of k-fold processing to maintain temporal relationships
4. The limited sample size for k-fold processing since shuffling is not possible
5. A nested k-fold process, as in the Alpaydin F-test

ME developers have and will continue to focus on ensuring that tabular and time series data are managed and evaluated appropriately, thus allowing users to use ME regardless of use case.

For more information on the differences between tabular and time series modeling, see this report [11] and this article [12].

Future

The developers intend for Model Evaluator to be a "living package" that grows over time. There is literally no limit to how the basic concepts presented in this paper and incorporated into ME could be extended over time. Here are some of the immediate opportunities that we are exploring now:

1. Add new evaluation methods, including Bayesian and Causal Inference
2. Reduce reliance on NHST methods, which entail several limitations to establishing robust statistical significance [17]
3. Allow users to select and add post-processing methods

Summary

The developers aim to establish ME as the premier tool for assessing statistical significance for machine learning. To do so, we will need the support of others, so we invite comments, criticisms, and contributors at (TBD). More broadly, we hope ME will encourage further analysis and development in this crucial area to benefit AI/ML practitioners and users.

References

[1] Avishek, Nag. 2022. *Bayesian Approach and Model Evaluation* [https://towardsdatascience.com/bayesian-approach-and-model-evaluation-371ad669cf2c]

[2] BayComp: A library for Bayesian comparison of classifiers [https://baycomp.readthedocs.io/en/latest/]

[3] B., Athina. 2020. *Statistical Tests and Their Importance* [https://link.medium.com/aaZWW8U8Rxb]

[4] Dienes, Z and N. Mclatchie. 2017. Four reasons to prefer Bayesian analyses over significance testing .... [https://link.springer.com/article/10.3758/s13423-017-1266-z]

[5] Dietterich, T. G. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms https://sci2s.ugr.es/keel/pdf/algorithm/articulo/dietterich1998.pdf

[6] Glen, Stephanie. McNemar Test Definition, Examples, Calculation [https://www.statisticshowto.com/mcnemar-test/]

[7] Greenglas. 2020. Can we talk about statistical significance using Bayesian Inference .... [https://stats.stackexchange.com/questions/560459/can-we-talk-about-statistical-significance-using-bayesian-inference]

[8] Kelter, R. 2020. Bayesian alternatives to null hypothesis significance testing in biomedical research: a non-technical introduction to Bayesian inference with JASP [https://pubmed.ncbi.nlm.nih.gov/32503439/]

[9] Kelter, R. 2020. Bayesian and frequentist testing for differences between two groups with parametric and nonparametric two-sample tests [https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wics.1523]

[10] Lacoste, Alexandre et. al. 2012. Bayesian Comparison of Machine Learning Algorithms on Single and Multiple Datasets *[http://proceedings.mlr.press/v22/lacoste12/lacoste12.pdf]*

[11] Madhana Bala, S. K. 2023. [https://wandb.ai/madhana/Time_Series/reports/Python-Time-Series-Forecasting-A-Practical-Approach--VmlldzoyODk4NjUz]

[12] Parmezan, A, V. Souza, G. Batista. 2019. Evaluation of statistical and machine learning models for time series prediction https://doi.org/10.1016/j.ins.2019.01.076

[13] Raschka, Sebastian. 2018. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning [https://arxiv.org/abs/1811.12808]

[14] Ross, Kevin. 2020. [https://bookdown.org/kevin_davisross/bayesian-reasoning-and-methods/model-comparison.html]

[15] Varoquaux, G. and O. Colliot. 2023. Evaluating machine learning models and their diagnostic value. https://hal.science/hal-03682454v4

[16] NIST ENGINEERING HANDBOOK: https://www.itl.nist.gov/div898/handbook/pri/section5/pri5322.htm

[17] Jeffrey A. Gliner, Nancy L. Leech, George A. Morgan, 2002. Problems With Null Hypothesis Significance Testing (NHST): What Do the Textbooks Say? https://www.andrews.edu/~rbailey/Chapter%20two/7217331.pdf

[18] Why is Model Evaluation Important in Machine Learning? Comet. November, 2022. https://www.comet.com/site/blog/why-is-model-evaluation-important-in-machine-learning/

[19] Jason Fernando. April 08, 2023. R-Squared: Definition, Calculation Formula, Uses, and Limitations. https://www.investopedia.com/terms/r/r-squared.asp

[20] David M. Lane. Difference between Two Means (Independent Groups). https://onlinestatbook.com/2/tests_of_means/difference_means.html

[21] Charles Zaiontz. Diebold-Mariano Test. https://real-statistics.com/time-series-analysis/forecasting-accuracy/diebold-mariano-test/

[22] Vishal, V. 2017. Boston Housing Dataset. Kaggle. https://www.kaggle.com/datasets/altavish/boston-housing-dataset

[23] Dablander, F. (2020, February 13). An Introduction to Causal Inference. https://doi.org/10.31234/osf.io/b3fkw