**Honest ML: A library for building confidence in statistical models**

# Summary

Machine learning metrics are built around the idea of training a model and then making out-of-sample predictions to test generalizability. There are a few standard methods; splitting the data into training and testing data and then predicting once on the testing or out of sample data. Using cross-validation to train on partitions of the data and then test by using one partition as the holdout and averaging the metric across all partitions. And finally, stratified partitioning splits the data subject to some condition, usually on the proportion of labels in the entire dataset. This paper will look at a library that implements a different method, training the model on many train-test splits and recording the out-of-sample error across these five hundred to more than a thousand splits. This creates higher confidence in the model and more closely simulates the likely scenarios you would find in the production setting, even with reasonably small datasets. Through this library, users can present statistical models based on confidence intervals to capture the uncertainty in inferences instead of point statistics for different machine learning models.

# Introduction

The idea of out-of-sample prediction is described in detail throughout the literature (Montgomery, 1991). The basic idea is to split the data into two groups, a training sample and a testing sample. Once the data is split, a statistical model is trained on the training sample. Then the trained model is used to predict the independent variable from the dependent variables (Kuhn & Johnson, 2013; Pawluszek-Filipiak & Borkowski, 2020) in the testing sample. Finally, a loss metric, like mean squared error, is used if it is a regression problem, or cross-entropy (Bickel & Doksum, 2015; James, 2021) is used for classification to compare the predicted dependent variable against the ground truth dependent variable. This method can be helpful as a first pass to assess model quality; however, it has many deficiencies (Doan et al., 2022; Salazar et al., 2022; Tan et al., 2021) since the data was only split once considering a classification problem, there may be issues such as:

1. Imbalance in the label classes in the training and testing data. This balance is not different from the entire data set, as well as the population data being modeled.
2. Concentration of independent variables caused by a specific exogenous effect (Edelkamp et al., 2021) in the training data and a different exogenous effect in the testing data.

If either of these conditions persists, our loss metric may record a far too optimistic or pessimistic view of how well the model performs. This, in turn, may have consequences for a whole host of things - failure to select the correct model, for instance, we may choose a logistic regression model (Gortmaker, 1994; Vittinghoff et al., 2012) when a decision tree model (de Ville, 2013; Shalev-Shwartz & Ben-David, 2013) is more appropriate. Or we may select the wrong hyperparameters for a given model class. A direct consequence of a flawed model is a poor inference which may have complex or impossible to recognize consequences (Chernozhukov et al., 2022; Kok, 2007; Marsili & Roudi, 2022; z\_ai, 2020). Therefore, it is of paramount importance that our models be 'honest' and the error well captured.

To deal with this failure to generalize from a single training and testing split, cross-validation (Arlot & Celisse, 2010; Kohavi, 1995) was created to increase the number of training and testing splits and then average the error metric or metrics. This works by creating several random partitions of the data and then treating one of the partitions as an out of the sample, while the rest are treated as in the sample. A model is trained on all in-sample predictions, and the out-of-sample is left for testing the model. The procedure is repeated for each partition used as an out-of-sample. Issues with choosing the optimum number of partitions, including multiple and separate partitions, may not generalize well in some cases; few partitions will produce the same problems as with a train-test split.

In theory, these methods described are inherently good approaches; the issues raised come down to how models are viewed and interpreted in practice. Therefore,  [honest\_ml](https://github.com/EricSchles/honest_ml) is a library to do many individual data splits, typically on the order of 500 to several thousand data splits. The idea is to iterate over the random seed used in a typical train-test split implementation. For this library, we use scikit-learn's implementation (Buitinck et al., 2013; Pedregosa et al., 2011), considered the gold standard by machine learning engineers. Doing so removes the need to consider how many partitions are required for a particular dataset. We also further decrease the possibility of a “lucky or unlucky” split in a train-test split. In addition, this implementation helps to identify the sensitivity of trained models to the data used in training the models with specific hyperparameters.

# Utilization

[honest\_ml](https://github.com/EricSchles/honest_ml) has an EvaluateModel class that allows users to pass in their classifier of choice, a target data set, a feature data set and the number of trials where each data split during a trial uses a different random seed. The relevant performance metrics are calculated for each train-test split. For example, in Figure 1, users can create an object of EvaluateModel. The performance metrics for each trial are saved after fitting the model.



Figure 1. Sample code for using the EvaluateModel class in honest\_ml

The [honest\_ml](https://github.com/EricSchles/honest_ml) library also have a visualization tool that allows users to view results of each trial relative to other trials stored in a user defined variable using the EvaluateModel class.

For example, using the model\_instances created above in the logistic regression model, users can compare metrics such as the precision, recall and f1-score for classification models. Figure 2 and Figure 3 shows the distribution of the precision and recall for 200 trials of the logistic regression model with two classes 0 and 1. Models that produce less normal distributions indicate a sensitivity of the model to the training data and provides users with a realistic expectation of the model in production than a point statistic would provide.

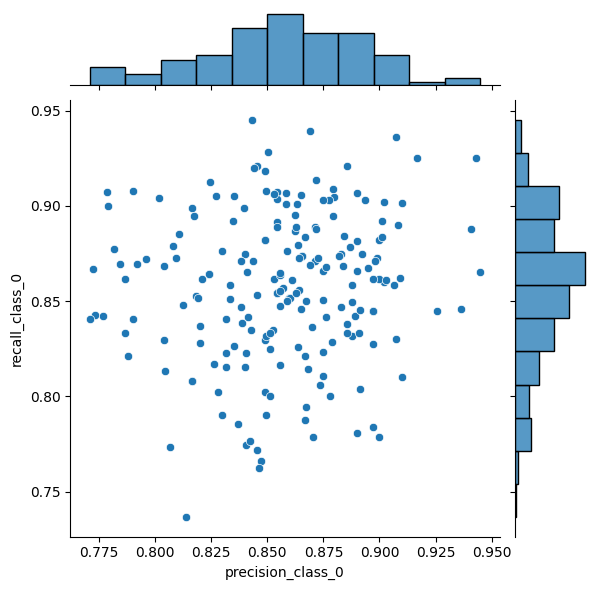


Figure . Comparison of the distribution of the precision and recall for different trials for the class 0

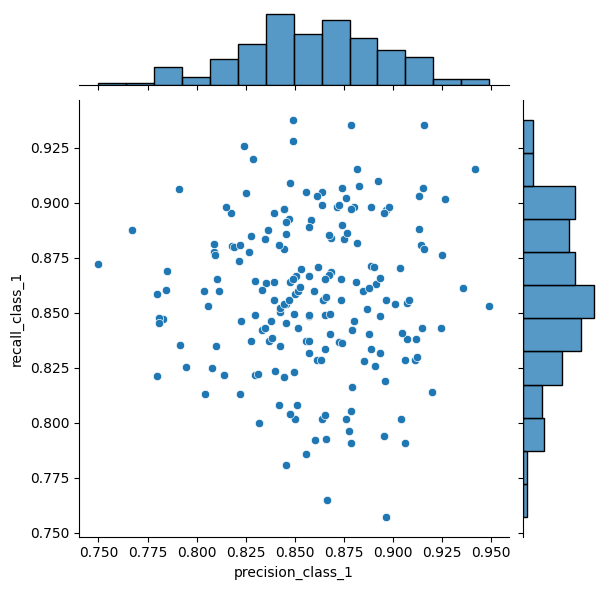


Figure . Sensitivity of class 1 to different trials using recall and precision distribution

# References

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*(none), 40-79. <https://doi.org/10.1214/09-SS054>

Bickel, P. J., & Doksum, K. A. (2015). *Mathematical statistics: Basic ideas and selected topics, second edition* (Vol. 1). <https://doi.org/10.1201/b18312>

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., & Grobler, J. (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*.

Chernozhukov, V., Newey, W. K., & Singh, R. (2022). Automatic Debiased Machine Learning of Causal and Structural Effects. *Econometrica*, *90*(3), 967-1027. <https://doi.org/10.3982/ECTA18515>

de Ville, B. (2013). Decision trees. *Wiley interdisciplinary reviews. Computational statistics*, *5*(6), 448-455. <https://doi.org/10.1002/wics.1278>

Doan, Q. H., Mai, S.-H., Do, Q. T., & Thai, D.-K. (2022). A cluster-based data splitting method for small sample and class imbalance problems in impact damage classification. *Applied soft computing*, *120*, 108628. <https://doi.org/10.1016/j.asoc.2022.108628>

Edelkamp, S., Möller, R., & Rueckert, E. (2021). *KI 2021: advances in artificial intelligence : 44th German Conference on AI, virtual event, September 27 - October 1, 2021 : proceedings*. Springer.

Gortmaker, S. L. (1994). Theory and methods -- Applied Logistic Regression by David W. Hosmer Jr and Stanley Lemeshow. In (Vol. 23, pp. 159). Washington: Sage Publications Ltd.

James, G. (2021). *An introduction to statistical learning : with applications in R* (2nd ed.). Springer.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai,

Kok, J. N. (2007). *Machine learning : ECML 2007 : 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007 : proceedings* (1st 2007. ed.). Springer. <https://doi.org/10.1007/978-3-540-74958-5>

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York. <https://doi.org/10.1007/978-1-4614-6849-3>

Marsili, M., & Roudi, Y. (2022). Quantifying relevance in learning and inference. *Physics reports*, *963*, 1-43. <https://doi.org/10.1016/j.physrep.2022.03.001>

Montgomery, D. C. (1991). Response surface methods and designs. *Design and analysis of experiments*.

Pawluszek-Filipiak, K., & Borkowski, A. (2020). On the Importance of Train–Test Split Ratio of Datasets in Automatic Landslide Detection by Supervised Classification. *Remote sensing (Basel, Switzerland)*, *12*(18), 3054. <https://doi.org/10.3390/rs12183054>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Salazar, J. J., Garland, L., Ochoa, J., & Pyrcz, M. J. (2022). Fair train-test split in machine learning: Mitigating spatial autocorrelation for improved prediction accuracy. *Journal of Petroleum Science and Engineering*, *209*, 109885. <https://doi.org/https://doi.org/10.1016/j.petrol.2021.109885>

Shalev-Shwartz, S., & Ben-David, S. (2013). *Understanding machine learning: From theory to algorithms* (Vol. 9781107057135). <https://doi.org/10.1017/CBO9781107298019>

Tan, J., Yang, J., Wu, S., Chen, G., & Zhao, J. (2021). A critical look at the current train/test split in machine learning.

Vittinghoff, E., Glidden, D. V., Shiboski, S. C., & McCulloch, C. E. (2012). *Regression Methods in Biostatistics Linear, Logistic, Survival, and Repeated Measures Models* (2nd 2012. ed.). Springer New York. <https://doi.org/10.1007/978-1-4614-1353-0>

z\_ai. (2020). *The Ultimate Guide to Debugging your Machine Learning models* [Article].