Better Data Splits for Machine Learning with astartes

JACKSON W. BURNS, Massachusetts Institute of Technology Center for Computational Science and Engineering

WILLIAM H. GREEN*, Massachusetts Institute of Technology Department of Chemical Engineering

Machine Learning (ML) has become an increasingly popular approach to accelerate traditional workflows. Critical to the use of ML is the process of splitting datasets into training, validation, and testing subsets that are used to develop and evaluate models. Common practice in the literature is to assign these subsets randomly. Although this approach is fast and efficient, it only measures a model's capacity to interpolate. Testing errors from random splits may be overly optimistic if given new data that is dissimilar to the scope of the training set; thus, there is a growing need to easily measure performance for extrapolation tasks. To address this issue, we report `astartes`, an open-source Python package that implements many similarity- and distance-based algorithms to partition data into more challenging splits. `astartes` operates on arbitrary vector inputs, so its principles and workflow are generalizable to any ML domain. `astartes` is available via the Python package managers `pip` and `conda` and is publicly hosted on GitHub (github.com/JacksonBurns/astartes).

Additional Key Words and Phrases: Python, machine learning, sampling, extrapolation, interpolation

ACM Reference Format:

Jackson W. Burns and William H. Green. 2023. Better Data Splits for Machine Learning with `astartes`. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.5281/zenodo.8155441

1 STATEMENT OF NEED

Machine Learning (ML) has catalyzed rapid progress across science, especially in chemical kinetics [6, 10], drug discovery [1, 13], materials science [12], and energy storage [5]. Researchers use data-driven methods to generate models that accelerate steps in traditional workflows within some acceptable error tolerance. To facilitate adoption of these models, researchers must critically think about several topics, such as comparing model performance to relevant baselines, operating on user-friendly inputs, and reporting performance on both interpolative and extrapolative tasks. `astartes` aims to make it straightforward for ML scientists and researchers to focus on rigorous hyperparameter optimization and accurate performance evaluation.

`astartes`' key function `train_val_test_split` returns splits for training, validation, and testing sets using an `sklearn`-like interface. These splits can then separately be used with any chosen ML model. This partitioning is crucial since best practices in data science dictate that avoiding data leakage and overfitting requires optimizing hyperparameters with a validation set and using a held-out test set to accurately measure performance on unseen data [3, 4, 7, 9, 11]. Unfortunately many published papers across domains mention training and testing sets but not validation sets, implying that they optimize the hyperparameters to the test set. This data leakage leads to overly optimistic results. However, for users interested in quickly obtaining preliminary results without using a validation set, `astartes` also implements an `sklearn`-compatible `train_test_split` function.

*advisor

Manuscript submitted to ACM

2 RELATED SOFTWARE AND CODE AVAILABILITY

In the ML space `astartes` functions as a drop-in replacement for the ubiquitous `train_test_split` from `scikit-learn` [8]. Transitioning existing code to use this new methodology is as simple as running `pip install astartes`, modifying an import statement at the top of the file, and then specifying an additional keyword parameter. `astartes` has been designed to allow for maximum interoperability with other packages, using few dependencies, supporting all platforms, and validated support for Python 3.7 through 3.11. Specific tutorials on this transition are provided in the online documentation for `astartes`, which is available on GitHub (github.com/JacksonBurns/astartes).

Here is an example workflow using texttttrain_test_split taken from the `scikit-learn` documentation [8]

```
import numpy as np
from sklearn.model_selection import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.33, random_state=42)
```

To switch to using `astartes`, `from sklearn.model_selection import train_test_split` becomes `from astartes import train_test_split` and the call to split the data is nearly identical and simple in the extensions that it provides:

```
import numpy as np
from astartes import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.33, sampler="kmeans", random_state=42)
```

With this small change, an extrapolative sampler based on k-means clustering will be used.

Inside cheminformatics, `astartes` makes use of all molecular featurization options implemented in `AIMSim` [2], which includes those from virtually all popular descriptor generation tools used in the cheminformatics field.

The codebase itself complies with pyOpenSci software standards, having a clearly defined contribution guideline and thorough, easily accessible documentation. `astartes` uses GitHub actions for Constant Integration testing including unit tests, functional tests, and regression tests. Test coverage is >99% and all proposed changes must maintain this standard as well as satisfy the regression tests.

REFERENCES

[1] Pauric Bannigan, Matteo Aldeghi, Zeqing Bao, Florian Häse, Alan Aspuru-Guzik, and Christine Allen. 2021. Machine Learning Directed Drug Formulation Development. *Advanced Drug Delivery Reviews* 175 (2021), 113806.

[2] Himaghna Bhattacharjee, Jackson Burns, and Dionisios G. Vlachos. 2023. AIMSim: An accessible cheminformatics platform for similarity operations on chemicals datasets. *Computer Physics Communications* 283 (2023), 108579. https://doi.org/10.1016/j.cpc.2022.108579

[3] Aurélien Géron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Inc.

[4] Chip Huyen. 2022. *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications.* O'Reilly Media, Inc.

[5] Swarn Jha, Matthew Yen, Yazmin Salinas, Evan Palmer, John Villafuerte, and Hong Liang. 2023. Learning-Assisted Materials Development and Device Management in Batteries and Supercapacitors: Performance Comparison and Challenges. *Journal of Materials Chemistry A* 11 (2023), 3904–3936.

[6] Evan Komp, Nida Janulaitis, and Stéphanie Valleau. 2022. Progress Towards Machine Learning Reaction Rate Constants. *Physical Chemistry Chemical Physics* 24 (2022), 2692–2705.

[7] Valliappa Lakshmanan, Sara Robinson, and Michael Munn. 2020. *Machine Learning Design Patterns: Solutions to Common Challenges in Data Preparation, Model Building, and MLOps.* O'Reilly Media, Inc.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[9] Bharath Ramsundar, Peter Eastman, Patrick Walters, and Vijay Pande. 2019. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More.* O'Reilly Media, Inc.

[10] Kevin A. Spiekermann, Lagnajit Pattanaik, and William H. Green. 2022. Fast Predictions of Reaction Barrier Heights: Toward Coupled-Cluster Accuracy. *The Journal of Physical Chemistry A* 126, 25 (2022), 3976–3986.

[11] Anthony Yu-Tung Wang, Ryan J. Murdock, Steven K. Kauwe, Anton O. Oliynyk, Aleksander Gurlo, Jakoah Brgoch, Kristin A. Persson, and Taylor D. Sparks. 2020. Machine Learning for Materials Scientists: An Introductory Guide Toward Best Practices. *Chemistry of Materials* 32, 12 (2020), 4954–4965.

[12] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. 2019. Machine Learning in Materials Science. *InfoMat* 1, 3 (2019), 338–358.

[13] Xin Yang, Yifei Wang, Ryan Byrne, Gisbert Schneider, and Shengyong Yang. 2019. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. *Chemical Reviews* 119, 18 (2019), 10520–10594.