PMLB v1.0: an open source dataset collection for benchmarking machine learning methods

This manuscript (permalink) was automatically generated from EpistasisLab/pmlb-manuscript@1b9b49b on September 11, 2020.

Authors

- **Trang T. Le**
  0000-0003-3737-6565 · trang1618 · trang1618
  Department of Biostatistics, Epidemiology and Informatics, Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104

- **William La Cava**
  0000-0002-1332-2960 · lacava · w_la_cava
  Department of Biostatistics, Epidemiology and Informatics, Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104

- **Joe Romano**

  None

- **Daniel Goldberg**

  None

- **Praneel Chakraborty**

  None

- **Natasha Ray**

  None

- **Weixuan Fu**
  0000-0002-6434-5468 · weixuanfu · weixuanfu
  Department of Biostatistics, Epidemiology and Informatics, Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104

- **Jason H. Moore**
  0000-0002-5015-1099 · EpistasisLab · moorejh
  Department of Biostatistics, Epidemiology and Informatics, Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104 · Funded by National Institutes of Health Grant Nos. LM010098 and AI116794.

Abstract

Introduction

Benchmarking is a standard practice to illustrate the strengths and weaknesses of algorithms regarding different problem characteristics. In machine learning (ML), benchmarking often involves assessing the performance of the ML models, namely how well they predict labels for new samples (supervised learning) or detect patterns among samples with no pre-existing labels (unsupervised learning) in a group of benchmark datasets. Compiled from a broad range of existing ML benchmark collection, the Penn Machine Learning Benchmark (PMLB) unified publicly available datasets from large repositories such as Kaggle and OpenML, enabling systematic assessment of different ML methods.

The first release of PMLB received overwhelmingly positive feedback from the ML community, reflecting the pressing need for a collection of standardized datasets to evaluate models. As the repository becomes more widely used, community members have requested new features such as additional information about the datasets as well as new functions to select datasets given specific criteria. In this paper, we reviewed existing functionality and presented new enhancements that help facilitate the user and contributor's frictionless interaction with the repository.

Methods

New datasets with rich metadata

Since its initial release, we have made major improvements in the collection of datasets as well as other helpful supporting features. We have redesigned the repository structure, and the collection now has benchmark datasets for regression problems (Fig. 1). To fulfill several users' request, each dataset also has a metadata.yaml file — an example of can be viewed here. As the name suggests, these files contain general information about the datasets themselves. Specifically, for each dataset, the metadata file includes the link to the original source of the dataset, a general description, the publication associated with the dataset generation, the type of problem (i.e., classification or regression), keywords (e.g., simulation, ecological, bioinformatics), and the features' description and their coding (e.g., 'non-promoter'= 0, 'promoter'= 1).

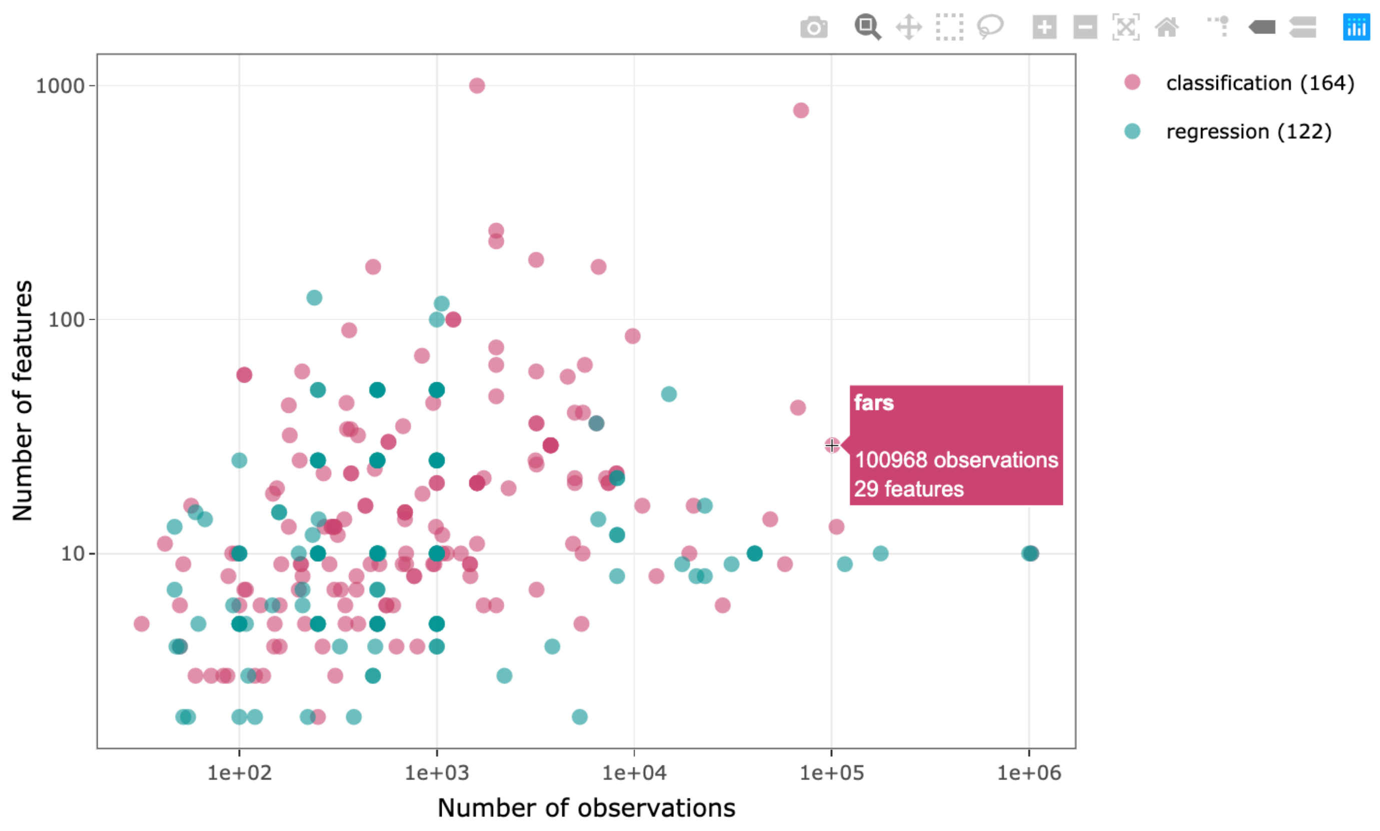
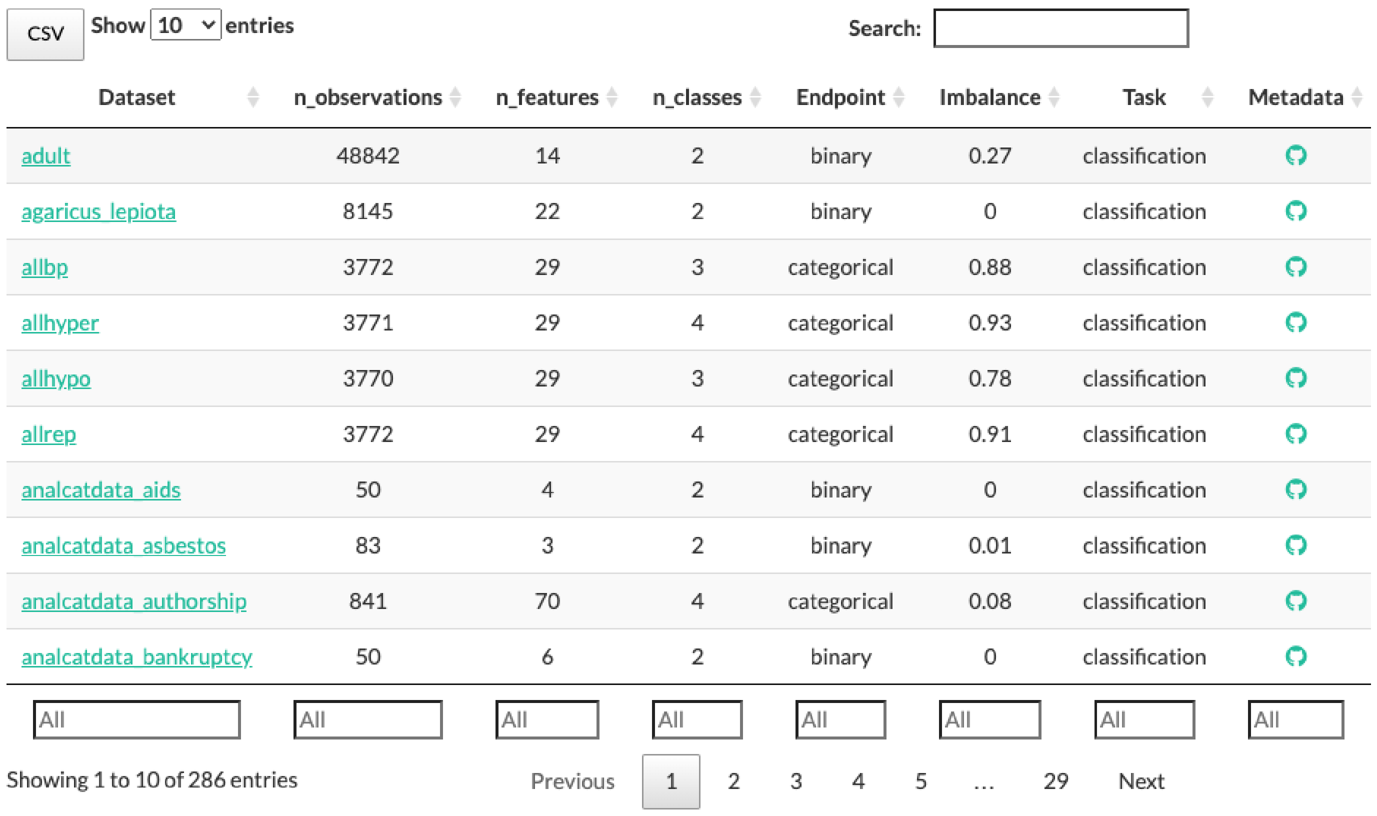


Figure 1: Characteristics of datasets in the PMLB collection

We are grateful for the open source contributors who have gradually increased the number of datasets with metadata. By carefully examining the data source and gather important information about the dataset, contributors have flagged serious issues with some datasets such as the incorrect column assigned as 'target' in the bupa dataset.

User-friendly interfaces

On the project PMLB home page, the user can now browse, sort, filter and search from a lookup table of datasets with summary statistics (Fig. 2). To select datasets with specific parameters, one can type in the box at the bottom of each numeric column in the format `low ... high`. For example, if the user wants to view all *classification* datasets with *80 to 100 observations*, they would select *classification* at the bottom of `Task` and type `80 ... 100` at the bottom of the `n_observations` column. The `CSV` button enables the download of the table with selected criteria.


CSV Show 10 entries Search:

Dataset	n_observations	n_features	n_classes	Endpoint	Imbalance	Task	Metadata
adult	48842	14	2	binary	0.27	classification	●
agaricus_lepiota	8145	22	2	binary	0	classification	●
allbp	3772	29	3	categorical	0.88	classification	●
allhyper	3771	29	4	categorical	0.93	classification	●
allhypo	3770	29	3	categorical	0.78	classification	●
allrep	3772	29	4	categorical	0.91	classification	●
analcatdata_aids	50	4	2	binary	0	classification	●
analcatdata_asbestos	83	3	2	binary	0.01	classification	●
analcatdata_authorship	841	70	4	categorical	0.08	classification	●
analcatdata_bankruptcy	50	6	2	binary	0	classification	●
All	All	All	All	All	All	All	All

Showing 1 to 10 of 286 entries Previous 1 2 3 4 5 … 29 Next

Figure 2: The user can browse, sort, filter and search the summary statistics table

On this main website, we also publish a detailed contribution guide with step-by-step instruction on how to get started. Our goal was to simplify the steps newcomers need to take to contribute. Specifically, we have automated many of these steps with continuous integration using the GitHub Actions service. When a new dataset is added, its summary statistics (e.g., number of observations, number of classes, etc.) is automatically computed, its profiling report is generated (see below), a corresponding metadata template is added to the dataset folder, and the list of dataset names is updated. Other checks included in the continuous integration workflow also help reduce both the reviewer and contributor's workload.

In addition to the Python library, we have integrated an R library – both can be simply installed with `pip install pmlb` or `install.packages('pmlb')`, respectively. This R library has been adapted from a separate repository that seemed to be unmaintained. However, because the original source code was released under a GPL-2 license, we were able to adapt the code to make it compatible with the new repository structure in this release and additional functionality. Is detailed vignettes also make PMLB a helpful resource for new users to begin testing their methods with benchmark datasets. These vignettes contain straightforward examples of how to automate the tedious task of comparing different ML methods on all the benchmark datasets based on specified metrics.

Pandas profiling reports

For each dataset, we use pandas profiling to provide a report for exploratory analysis. In addition to descriptive statistics of the features as provided by `pandas.describe` (Python) or `skimr::skim` (R), pandas profiling gives a more extensive exploration of the dataset such as correlation structure among the features and flaggings of duplicate rows. Browsing a report allows a user or contributor to easily assess the dataset quality and make necessary changes. For example, if a feature is flagged as having a unique value for each row in the report, it is likely that this feature is an observation identifier and should be removed from the dataset.

The profiling reports can be accessed by clicking on the dataset name in the data table or the data point in the interactive chart on the home page. Alternatively, all the reports can be viewed on the repository's gh-pages branch.

Efficiency

We have significantly reduced the repository size and started to track all data files with Git Large File Storage for efficient cloning of the repository. With the Large File Storage service, we now store large files on the GitHub.com remote server (with no limits on data storage) and include text pointers to these files in our repository. Users who want to interact with the entire repository on their local machine only need git LFS installed and set up for their user account or download the zip file from GitHub.

New functionality

PMLB now includes original data rows with missing data (i.e., NA). The core function of PMLB, fetch_data(), retains previous behavior (`dropna=True`) by default, which excludes all rows with missing data. However, if the user chooses to treat the missing values differently, they can use fetch_data() with the option `dropna=False` to obtain the original dataset and apply their own removal or imputation method.

Defining the neighborhood to be the datasets' metadata/characteristics space, we also enabled the option to select the nearest PMLB datasets given a data frame. This functionality would be helpful for users who would like to find PMLB datasets with similar characteristics to their own to make inference on their dataset, e.g., where to start the hyperparameter search. An API reference that details the user-facing functions and variables within the PMLB python library is included in the PMLB page.

Results and Discussion

References