

giordamaug / SVElearn

Code Issues Pull requests Actions Projects Security Insights Settings



Machine learning methods developed at ICAR-CNR

GPL-3.0 license

0 stars 0 forks 3 watching 1 Branch 6 Tags Activity

Public repository

1 Branch 6 Tags Go to file Go to file Code

giordamaug Delete build/lib/svelearn/validation/crossvalidate.py

ebb3c4a · last week

examples	Delete examples/cv_iris.ipynb	last week
images	Delete images/prova.txt	last week
svelearn	upd	last week
.gitattributes	Initial commit	3 weeks ago
.gitignore	upd	2 weeks ago
LICENSE	Initial commit	3 weeks ago
README.md	Update README.md	last week
setup.py	upd	last week

README License

SVElearn

Machine learning methods for unbalanced datasets

Authors

- Maurizio Giordano and Ilaria Granata
- High Performance Computing and Networking (ICAR), Italian National Council of Research (CNR)

Documentation

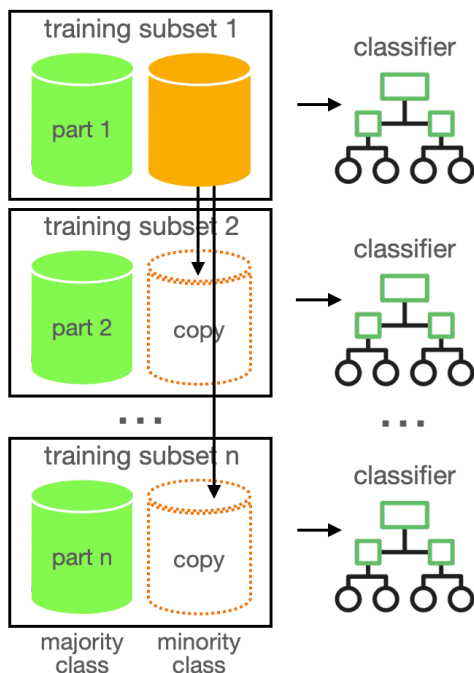
This package is a library of machine learning methods developed by ICAR-CNR for efficient prediction models in unbalanced data domains.

The Splitting Voting Ensemble

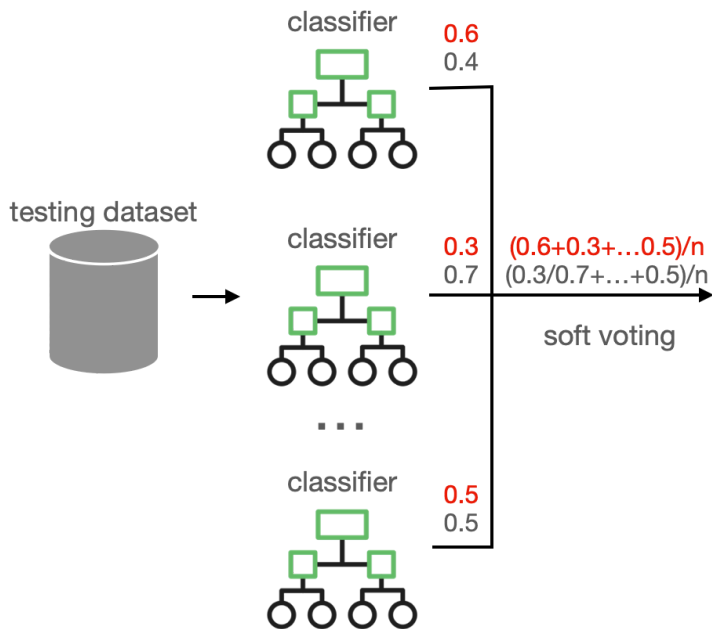
The Splitting Voting Ensemble (SVE) is a meta-model designed to address classification task on unbalanced machine learning datasets.

SVE can be considered a meta-learning algorithm since it uses another learning method as a base model for all members of the ensemble to combine their predictions. This algorithm was designed and developed to address binary and multiclass classification tasks in data domains characterized by strong unbalancing of classes, such as Cybersecurity, Bionformatics, etc.

Before training, the method partitions the set of majority class samples into n parts, and it trains each classifier on a subset of training data composed of one of these parts along with the entire set of minority class training samples. In multiclass scenarios, the partition ratio considered is that between the samples of the majority class and the second majority class: only the samples of the majority classes are partitioned, and each partition associated with a duplicate of the samples of the remaining classes.



During testing on unseen data, each classifier of the ensemble produces a probability for the label prediction; we compute the final probability response of the ensemble as the average of the probabilities of the n voting classifiers. The number n of classifiers is automatically determined by the algorithm according to the class distribution in training data, or user-specified as an input parameter ('`n_voters`').



Credits

The SVElearn package was developed by High Performance Computing and Networking Institute of National Research Council of Italy (ICAR-CNR). This software is released under the GNU Licence (v.3)

Cite

If you use want to reference this software, please use the DOI: doi/10.5281/zenodo.10964743

DOI [10.5281/zenodo.12699488](https://doi.org/10.5281/zenodo.12699488)


If you want to cite the work in which this software was first used and described, please cite the following article:

```
@article {Granata2024.04.16.589691,
  author = {Ilenia Granata and Lucia Maddalena and Mario Manzo and Mario
Rosario Guarracino and Maurizio Giordano},
  title = {HELP: A computational framework for labelling and predicting
human context-specific essential genes},
  elocation-id = {2024.04.16.589691},
  year = {2024},
  doi = {10.1101/2024.04.16.589691},
  publisher = {Cold Spring Harbor Laboratory},
  URL =
{https://www.biorxiv.org/content/early/2024/04/20/2024.04.16.589691},
  eprint =
```





Releases 2

 **Machine learning methods for unbalanced datasets** Latest
last week

[+ 1 release](#)

Packages


No packages published
[Publish your first package](#)


Languages




Suggested workflows

Based on your tech stack

 **Python Package using Anaconda** Configure
Create and test a Python package on multiple Python versions using Anaconda for package management.

 **Pylint** Configure
Lint a Python application with pylint.

 **SLSA Generic generator** Configure
Generate SLSA3 provenance for your existing release workflows

[More workflows](#)

[Dismiss suggestions](#)