
Continues Integration (CI)

— 02476 Machine Learning Operations —
Nicki Skafte Detlefsen

Why you should care about today

Two years ago the day before this lecture, the internet went down for a couple of hours because someone f..ked up their CI.

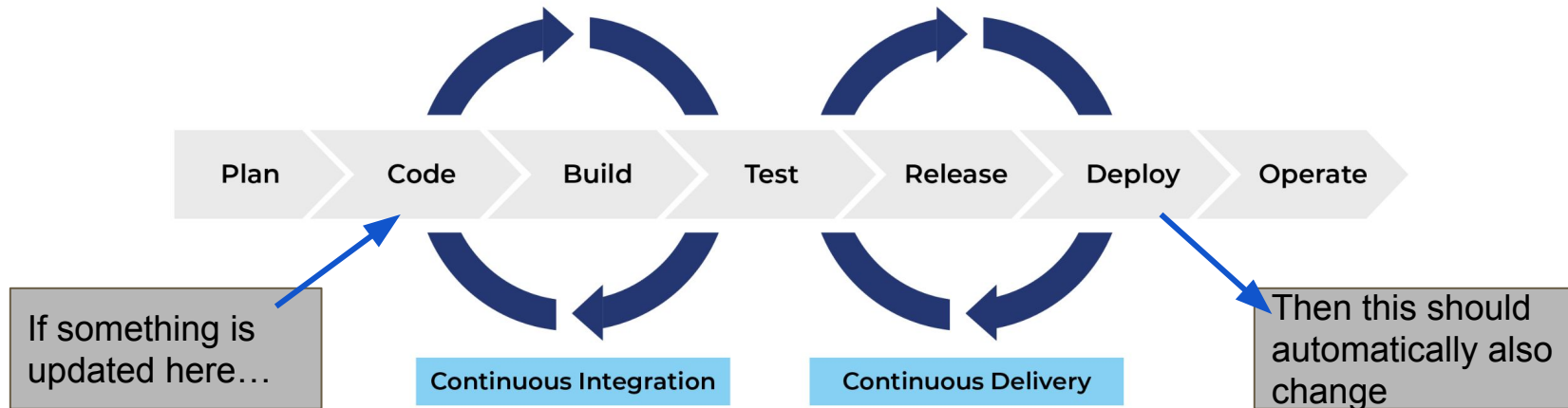
Dev at Fastly : I'll just push this small change to production

Dev at Fastly 2 seconds later:



Continues X

Term refers to a set of **software practises** for **automating** tedious tasks and make sure changes in a pipeline are **continuously propagated** through the pipeline.



CI vs CD vs CML

Continuous Integration

Core task:

How to automatically secure that code does not break during development?

App independent concept

Continuous Deployment

Core task:

How to get your code/ application to the end user automatically? + monitor life cycle

App dependent concept

Continuous ML

Core task:

How to automatic retrain machine learning models when data changes?

Specific to ML applications



- ML Model Deployment
- CI/CD Pipelines
- Monitoring & Triggering

This lecture: CI

Core task:

How to automatically secure that code does not break during development?

3 steps to do this:

1. Use version control:
 - Frequently committing code to shared repository
2. Write (unit-)test for your code
 - Should capture unwanted bugs in your code
3. Automate build + test
 - Automatic run test so code cannot be merged without working

Contributors 169

Languages



TorchMetrics

Machine learning metrics for distributed, scalable PyTorch applications.

[What is Torchmetrics](#) • [Implementing a metric](#) • [Built-in metrics](#) • [Docs](#) • [Community](#) • [License](#)

python 3.7 | 3.8 | 3.9 | 3.10 pypi package 0.10.3 downloads 28M conda v0.10.3 downloads 585k License Apache 2.0

CI testing - complete Azure Pipelines codecov 39%

slack chat docs passing DOI 10.5281/zenodo.5844769 JOSS 10.21105/joss.04101 pre-commit.ci passed

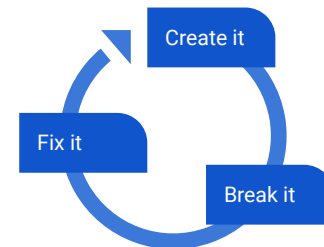
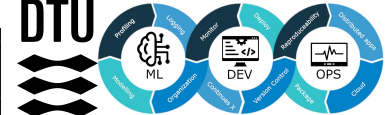
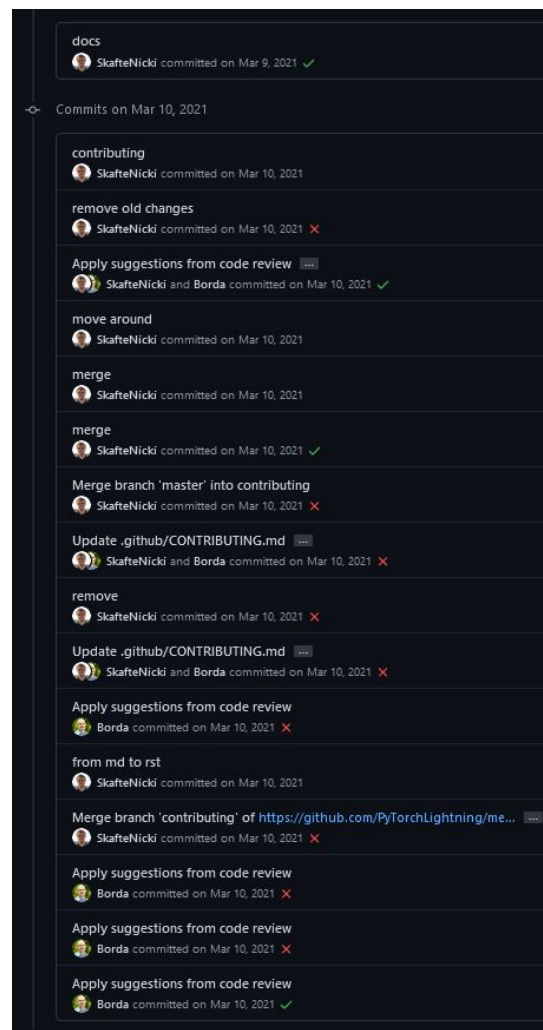
CI step 1: version control

Use version control

- Code changes are tracked
- Branches for parallel work

Commit frequently

- Catch errors sooner than later
- Merging can be done automatically



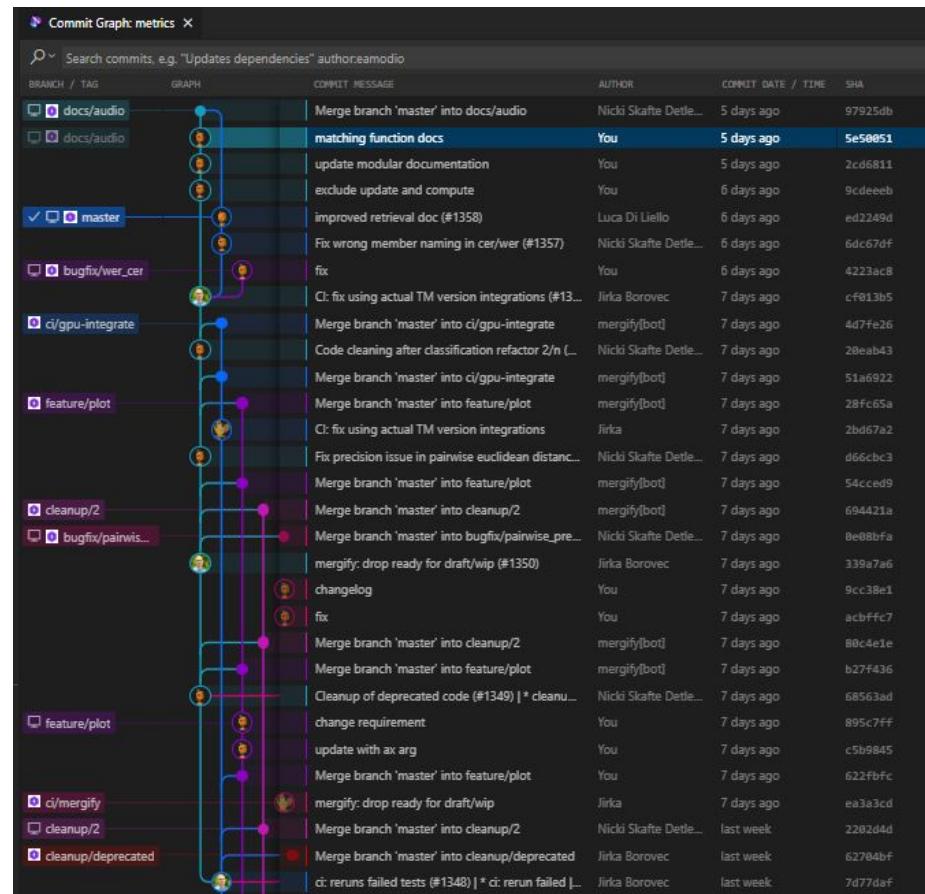
Ci step 1: Use branches

Parallel workflow

Experimental features/changes are kept away from master.

Recommend extensions for VScode:

- [Gitlens](#) or [GitGraph](#)
- [Github PR and issues](#)



CI step 1: version control

Lightning-AI / metrics
1. Find PR

Added MinkowskiDistance support #1362

2. Check changed files

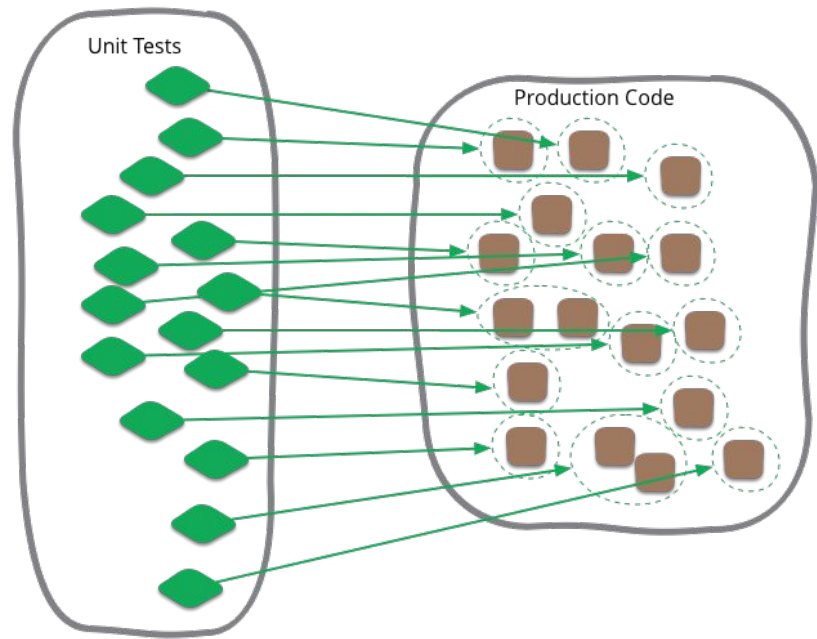
4. Send review

3. Make one or more comments

CI step 2: write tests

Test are the cornerstones of CI.

- In particular, *unit tests* are important.
- A single unittest, tests a small part of your code
- By testing code in small pieces, bugs are easier to find



CI step 2: write tests

By python convention
source code should be
in a **src** folder

master	metrics / src / torchmetrics /	Go to file	Add file	...
SkaftNicki	Fix wrong member naming in cer/wer (#1357)	6 days ago	History	
..				
audio	CI: re-use checks (#1261)	2 months ago		
classification	Code cleaning after classification refactor 2/n (#1252)	7 days ago		
detection	Revert mAP matching vectorization (#1327)	19 days ago		
functional	Code cleaning after classification refactor 2/n (#1252)	7 days ago		
image	Add normalize argument to certain image metrics (#1246)	21 days ago		
multimodal	Add CLIP score (#1314)	11 days ago		
nominal	Add Thiel's U Statistic (Uncertainty) Metric (#1337)	10 days ago		
regression	Add LogCosh Error (#1316)	21 days ago		
retrieval	CI: re-use checks (#1261)	2 months ago		
text	Fix wrong member naming in cer/wer (#1357)	6 days ago		
utilities	Code cleaning after classification refactor 2/n (#1252)	7 days ago		
wrappers	Code cleaning after classification refactor 2/n (#1252)	7 days ago		
__about__.py	docs: Update the number of metrics + covered domains (#1344)	11 days ago		
__init__.py	Exact match multiclass (#1343)	8 days ago		
aggregation.py	move: torchmetrics >> src/	6 months ago		
collections.py	Code cleaning after classification refactor 2/n (#1252)	7 days ago		
metric.py	Cleanup of deprecated code (#1349)	7 days ago		
py.typed	move: torchmetrics >> src/	6 months ago		

CI step 2: write tests

(Near) One to one match
src folder structure

In total 17727 tests

master metrics / tests / unittests /

Go to file Add file ...

6 authors Code cleaning after classification refactor 2/n (#1252) 20eab43 7 days ago History

..		
audio	Set minimum pytorch version to 1.8 + cleanup (#1263)	2 months ago
bases	Code cleaning after classification refactor 2/n (#1252)	7 days ago
classification	Cleanup of deprecated code (#1349)	7 days ago
detection	Revert mAP matching vectorization (#1327)	19 days ago
helpers	Add CLIP score (#1314)	11 days ago
image	Add normalize argument to certain image metrics (#1246)	21 days ago
multimodal	Add CLIP score (#1314)	11 days ago
nominal	Add Thiel's U Statistic (Uncertainty) Metric (#1337)	10 days ago
pairwise	Fix precision issue in pairwise euclidean distance (#1352)	7 days ago
regression	Add LogCosh Error (#1316)	21 days ago
retrieval	[Refactor] Classification 1/n (#1054)	3 months ago
text	Bugfix: Update message regarding connection issues to the HF hub (#1141)	5 months ago
utilities	Set minimum pytorch version to 1.8 + cleanup (#1263)	2 months ago
wrappers	Code cleaning after classification refactor 2/n (#1252)	7 days ago
init.py	rename tests/ (#1091)	6 months ago

CI step 2: write tests

In python, we recommend using the **pytest** framework.

Test are simple functions that start with *test_* and uses *assert*

```
import torch
from torch.nn.functional import mse_loss

def test_mse_loss_zeros():
    # (0 - 0)**2 = 0
    assert mse_loss(torch.zeros(1,), torch.zeros(1,)) == 0

def test_mse_loss_ones():
    # (1 - 0)**2 = 1
    assert mse_loss(torch.ones(1,), torch.zeros(1,)) == 0
```

CI step 2: write tests

Test can be simple...

```
def test_warning_on_nan(tmpdir):  
    preds = torch.randint(3, size=(20, ))  
    target = torch.randint(3, size=(20, ))  
  
    with pytest.warns(  
        UserWarning,  
        match='.* nan values found in confusion matrix have been replaced with zeros.',  
    ):  
        confusion_matrix(preds, target, num_classes=5, normalize='true')
```

CI step 2: write tests

Test can be simple...

```
def test_warning_on_nan(tmpdir):
    preds = torch.randint(3, size=(20,))
    target = torch.randint(3, size=(20,))

    with pytest.warns(
        UserWarning,
        match='.* nan values found in confusion matrix have been replaced with zeros.',
    ):
        confusion_matrix(preds, target, num_classes=5, normalize='true')
```

Or complicated

```
@pytest.mark.parametrize("normalize", ['true', 'pred', 'all', None])
@pytest.mark.parametrize(
    "preds, target, sk_metric, num_classes, multilabel",
    [
        (_input_binary_prob.preds, _input_binary_prob.target, _sk_cm_binary_prob, 2, False),
        (_input_binary_logits.preds, _input_binary_logits.target, _sk_cm_binary_prob, 2, False),
        (_input_binary.preds, _input_binary.target, _sk_cm_binary, 2, False),
        (_input_mlb_prob.preds, _input_mlb_prob.target, _sk_cm_multilabel_prob, NUM_CLASSES, True),
        (_input_mlb_logits.preds, _input_mlb_logits.target, _sk_cm_multilabel_prob, NUM_CLASSES, True),
        (_input_mlb.preds, _input_mlb.target, _sk_cm_multilabel, NUM_CLASSES, True),
        (_input_mcls_prob.preds, _input_mcls_prob.target, _sk_cm_multiclass_prob, NUM_CLASSES, False),
        (_input_mcls_logits.preds, _input_mcls_logits.target, _sk_cm_multiclass_prob, NUM_CLASSES, False),
        (_input_mcls.preds, _input_mcls.target, _sk_cm_multiclass, NUM_CLASSES, False),
        (_input_mdmc_prob.preds, _input_mdmc_prob.target, _sk_cm_multidim_multiclass_prob, NUM_CLASSES, False),
        (_input_mdmc.preds, _input_mdmc.target, _sk_cm_multidim_multiclass, NUM_CLASSES, False)]
)

class TestConfusionMatrix(MetricTester):

    @pytest.mark.parametrize("ddp", [True, False])
    @pytest.mark.parametrize("dist_sync_on_step", [True, False])
    def test_confusion_matrix(
        self, normalize, preds, target, sk_metric, num_classes, multilabel, ddp, dist_sync_on_step
    ):
        self.run_class_metric_test(
            ddp=ddp,
            preds=preds,
            target=target,
            metric_class=ConfusionMatrix,
            sk_metric=partial(sk_metric, normalize=normalize),
            dist_sync_on_step=dist_sync_on_step,
            metric_args={
                "num_classes": num_classes,
                "threshold": THRESHOLD,
                "normalize": normalize,
                "multilabel": multilabel
            }
        )
```

Parametrize is powerful:
4 x 11 x 2 x 2 = 176 tests!

CI step 2: execute locally

```
(lightning) C:\Users\nsde\Documents\metrics>pytest tests\unittests\regression\test_mean_error.py
===== test session starts =====
platform win32 -- Python 3.8.13, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: C:\Users\nsde\Documents\metrics, configfile: setup.cfg
plugins: cov-4.0.0, doctestplus-0.12.1, timeout-2.1.0
collected 116 items

tests\unittests\regression\test_mean_error.py sssssssssssss.....ssssssssssssss.....xxxxxxx

===== warnings summary =====
..\Anaconda3\envs\lightning\lib\site-packages\_pytest\config\_init_.py:1183
C:\Users\nsde\Anaconda3\envs\lightning\lib\site-packages\_pytest\config\_init_.py:1183: PytestDeprecationWarning: The --strict option is deprecated, use --strict-markers instead.
  self.issue_config_time_warning(
-- Docs: https://docs.pytest.org/en/stable/warnings.html
===== 80 passed, 28 skipped, 8 xfailed, 1 warning in 16.44s =====
```

- Test passed
- F Test failed
- S Test skipped (pytest.skipif, pytest.skip)
- X Test was expected to fail (pytest.xfail)

Do you remember to do this after each commit?
Lets automate doing it instead

CI step 3: Automating stuff

What can be automated: EVERYTHING

- Unit testing
- Integration testing
- Documentation creation
- Linters (style formatting)
- Security checks
- Code coverage
- Custom checks...

Only your imagination is the limit

CI step 3: Setup on your side

To automate testing we need to wrap into a package: create `setup.py`

Allows easy install

`python setup.py install` or `python setup.py devel`

or if uploaded to pip: `pip install my_package`

10 lines (9 sloc) | 362 Bytes

Raw

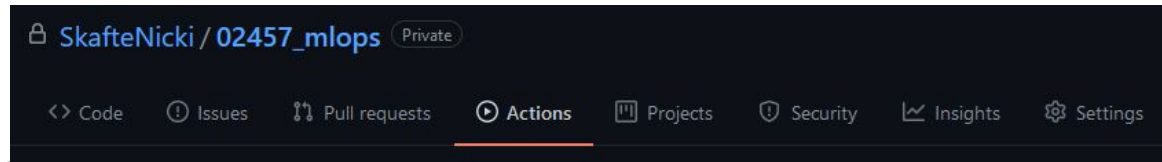
Blame



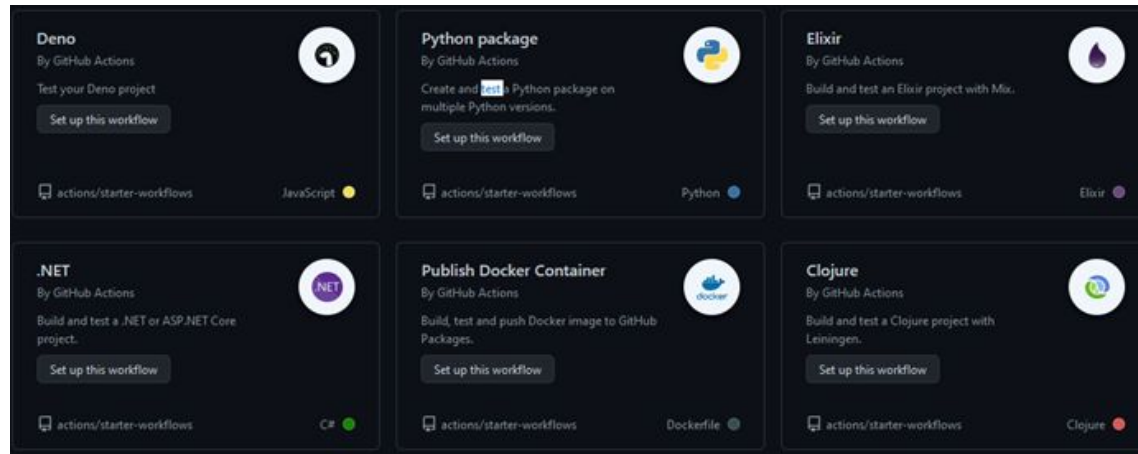
```
1 from setuptools import find_packages, setup
2
3 setup(
4     name='src',
5     packages=find_packages(),
6     version='0.1.0',
7     description='{{ cookiecutter.description }}',
8     author='{{ cookiecutter.author_name }}',
9     license='%(if cookiecutter.open_source_license == "MIT" %)MIT%(elif cookiecutter.open_source_license == "BSD-3-Clause" %)BSD-3%(endif %)',
10 )
```

CI step 3: Github actions

Built-in CI for github. Free 2,000 automation minutes/month (public repository)



Many ready to go workflows



CI step 3: Workflow files

Workflow files are a set of instructions that should be executed on a virtual machine

You can have many workflow files

When tests should be triggered

Define OS + python

Setup python

Install dependencies and package

Check formatting

Run tests

```

1  name: Python package
2
3  on:
4    push:
5      branches: [ main ]
6    pull_request:
7      branches: [ main ]
8
9  jobs:
10   build:
11
12     runs-on: ubuntu-latest
13     strategy:
14       matrix:
15         python-version: ["3.7", "3.8", "3.9", "3.10"]
16
17     steps:
18       - uses: actions/checkout@v3
19       - name: Set up Python ${ matrix.python-version }
20         uses: actions/setup-python@v4
21         with:
22           python-version: ${ matrix.python-version }
23       - name: Install dependencies
24         run: |
25           python -m pip install --upgrade pip
26           pip install flake8 pytest
27           pip install -r requirements.txt
28           python setup.py install
29       - name: Lint with flake8
30         run: |
31           flake8 src/
32       - name: Test with pytest
33         run: |
34           pytest tests/
35

```

CI step 3: Workflow files

master metrics / .github / workflows /

Borda ci: reruns failed tests (#1348) 7d77daf 7 days ago History

File	Description	Time
ci-checks.yml	CI: merge conda and full tests (#1324)	18 days ago
ci-docker.yml	ci: reruns failed tests (#1348)	7 days ago
ci-integrate.yml	ci: reruns failed tests (#1348)	7 days ago
ci-tests-full.yml	ci: reruns failed tests (#1348)	7 days ago
docs-check.yml	Add CLIP score (#1314)	11 days ago
focus-diff.yml	Bump actions/setup-python from 2 to 4 (#1169)	4 months ago
greetings.yml	Initial commit	2 years ago
publish-docker.yml	Bump docker/login-action fr	
publish-pypi.yml	Bump pypa/gh-action-pypi-p	

```
- name: Install dependencies
  run: |
    python -m pip install --upgrade --user pip
    pip install --requirement ./requirements.txt --find-links https://download.pytorch.org/whl/cpu/torch_stable.html
    pip install "pytest>6.0" "pytest-cov>2.10" --upgrade-strategy only-if-needed
    python --version
    pip --version
    pip list
  shell: bash

- name: Test Package [only]
  run: |
    # NOTE: run coverage on tests does not propagate faler status for Win, https://github.com/nedbat/coveragepy/issues/1003
    python -m pytest torchmetrics -v --cov=torchmetrics --junitxml=junit/test-results-${{ runner.os }}-${{ matrix.python-version }}.xml
```

CI step 3: Workflow files

Mergify

Summary

WIP

WIP

Docs check
on: pull_request

test-docs

make-docs

paper-JOSS

paper-cite

CI integrations
on: pull_request

pytest (ubuntu-20.04, 3.7, oldest)

pytest (ubuntu-20.04, 3.10, latest)

pytest (macOS-11, 3.7, oldest)

pytest (macOS-11, 3.10, latest)

pytest (windows-2022, 3.7, olde...

pytest (windows-2022, 3.10, lat...

pytest (3.10, latest, ubuntu-22.04)

pytest (3.10, latest, macOS-12)

CI testing - complete
on: pull_request

check-diff / eval-diff

pytest (ubuntu-20.04, 3.8, 1.9.1)

pytest (ubuntu-20.04, 3.8, 1.10.2)

pytest (ubuntu-20.04, 3.8, 1.11.0)

pytest (ubuntu-20.04, 3.8, 1.12.1)

pytest (ubuntu-22.04, 3.8, 1.13.0)

pytest (ubuntu-22.04, 3.10, 1.13...

pytest (macOS-11, 3.8, 1.13.0)

pytest (macOS-11, 3.9, 1.13.0)

pytest (windows-2022, 3.8, 1.13.0)

pytest (windows-2022, 3.9, 1.13.0)

pytest (ubuntu-20.04, 3.7, 1.8.1,...

pytest (ubuntu-20.04, 3.8, 1.8.1,...

pytest (macOS-11, 3.7, 1.8.1, old...

pytest (macOS-11, 3.8, 1.8.1, old...

pytest (windows-2019, 3.7, 1.8.1...

pytest (windows-2019, 3.8, 1.8.1...

General checks
on: pull_request

check-code / mypy

check-code / pre-commit

check-schema / schema

check-package / pkg-check (ub...

check-package / pkg-check (ub...

check-package / pkg-check (ma...

check-package / pkg-check (ma...

check-package / pkg-check (wi...

check-package / pkg-check (wi...

Azure Pipelines

Lightning-AI.metrics Re-run

Lightning-AI.metrics (pytest PyT... Re-run

Lightning-AI.metrics (pytest PyT... Re-run

43 checks in total

Test combination of

- Hardware setup
- Operating system
- Python version
- Dependencies

Runs unit tests, docs, coverage, linting, package installer etc.

CI step 3: Code is checked before merge

Protection rules:

- All/some tests should pass
- x developers should approve
- Comments should be taken care of

The screenshot shows a GitHub pull request interface with the following elements:

- Review required:** A red error message stating "At least 2 approving reviews are required by reviewers with write access. [Learn more.](#)" with a link to "Show all reviewers".
- 1 approval:** A green checkmark indicating one approval.
- 4 pending reviewers:** A red icon indicating four pending reviewers.
- No unresolved conversations:** A green checkmark indicating no unresolved conversations, with a link to "View".
- Some checks were not successful:** A red error message stating "4 failing and 43 successful checks" with a link to "Hide all checks".
- Check list:** A list of checks with their status and details:
 - Docs check / test-docs (pull_request):** Failing after 1m. Status: Required. Link: Details.
 - CI integrations / pytest (ubuntu-20.04, 3.7, oldest) (pull_request):** Successful in 1m. Status: Required. Link: Details.
 - CI integrations / pytest (ubuntu-20.04, 3.10, latest) (pull_request):** Successful in 1m. Link: Details.
 - CI integrations / pytest (macOS-11, 3.7, oldest) (pull_request):** Successful in 4m. Link: Details.
 - CI integrations / pytest (macOS-11, 3.10, latest) (pull_request):** Successful in 2m. Link: Details.
 - CI integrations / pytest (windows-2022, 3.7, oldest) (pull_request):** Successful in 7m. Link: Details.
- Merging is blocked:** A red error message stating "Merging can be performed automatically with 2 approving reviews."
- Enable auto-merge (squash):** A dropdown menu with the option "Automatically merge when all requirements are met. [Learn more](#)".

CI step 3: Automate tedious tasks with bots

```

[ ] and others added 5 commits 7 days ago
- [ ] Update test_auc.py Verified 674e7ea
- [P] [pre-commit.ci] auto fixes from pre-commit.com hooks 07a4b85
- [ ] Update test_auc.py Verified 85e4458
- [P] [pre-commit.ci] auto fixes from pre-commit.com hooks 0a94ba5
- [ ] Update test_auc.py Verified 0527efd
  
```

```

4 tests/classification/test_auc.py
@@ -93,9 +93,9 @@ def test_auc_differentiability(self, x, y, reorder):
93 def test_auc(x, y, expected, unsqueeze_x, unsqueeze_y):
94     if unsqueeze_x:
95         x = x.unsqueeze(-1)
96 -
97     if unsqueeze_y:
98         y = y.unsqueeze(-1)
99 -
100     # Test Area Under Curve (AUC) computation
101     assert auc(tensor(x), tensor(y), reorder=True) == expected
  
```


CI summary

1. Use version control



2. Write (unit-)test for your code



3. Automate build + test



Meme of the day



Gabriele Petronella
@gastro27

So this just happened:

- a bot found a vulnerability in a dependency
- a bot sent a PR to fix it
- the CI verified the PR
- a bot merged it
- a bot celebrated the merge with a GIF

