Continues Integration (CI)

02457 Machine Learning Operations

Nicki Skafte Detlefsen,

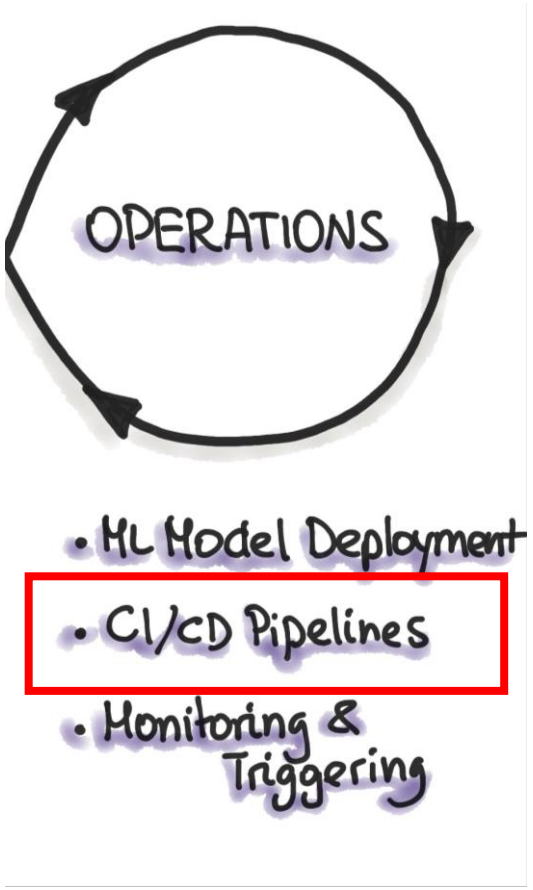
Postdoc

DTU Compute

What is continues integration?

MLOps

- Software practise
  - Frequitly commit code to shared reposatory
  - By commit sooner than later, errors are captured early
  - Make merging easier
  - Automate build + test
- App independent

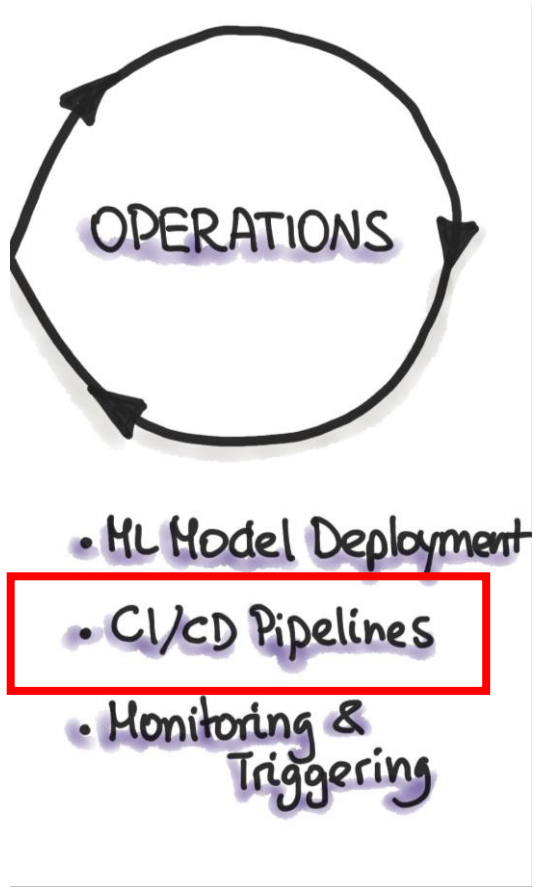


What is continues deployment?

MLOps

- How to get your code to the user
  - 3 phases: Testing, staging and deploying
- App dependent

Note: Not covered in this course



What should you know about CI?

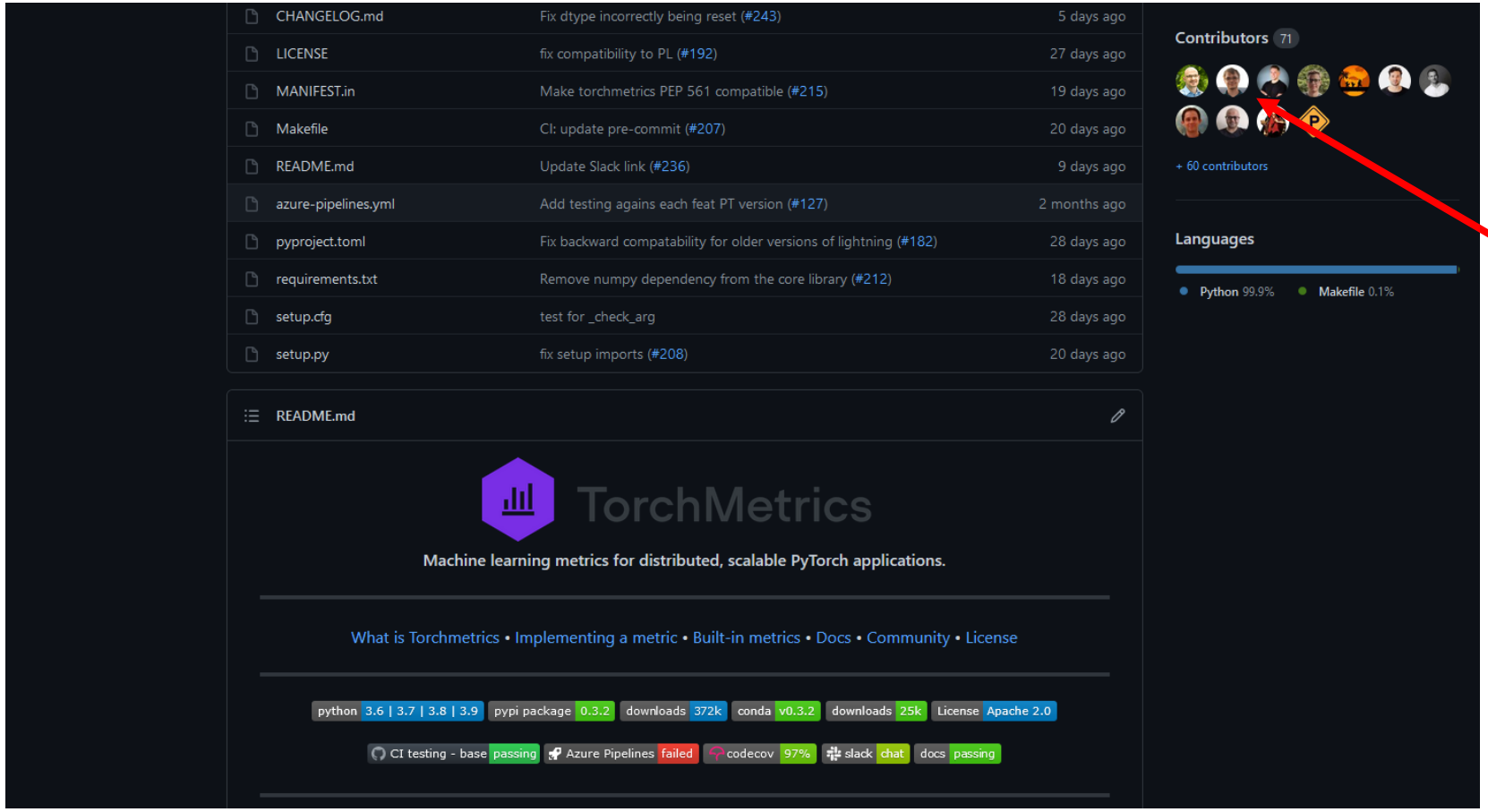
- CI is one of those topics that are best though as learning-by-doing
- If you understand how to use git, the rest is basically googling stuff

Lets look at a practical axample

A small case study

MLOps

All the metric in the world – now in pytorch

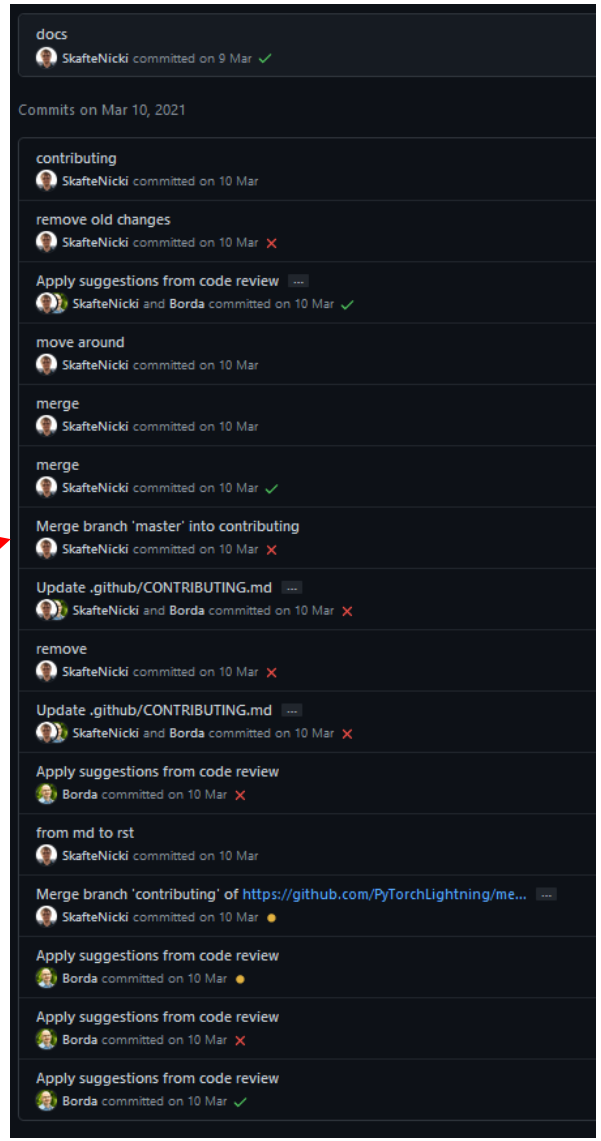


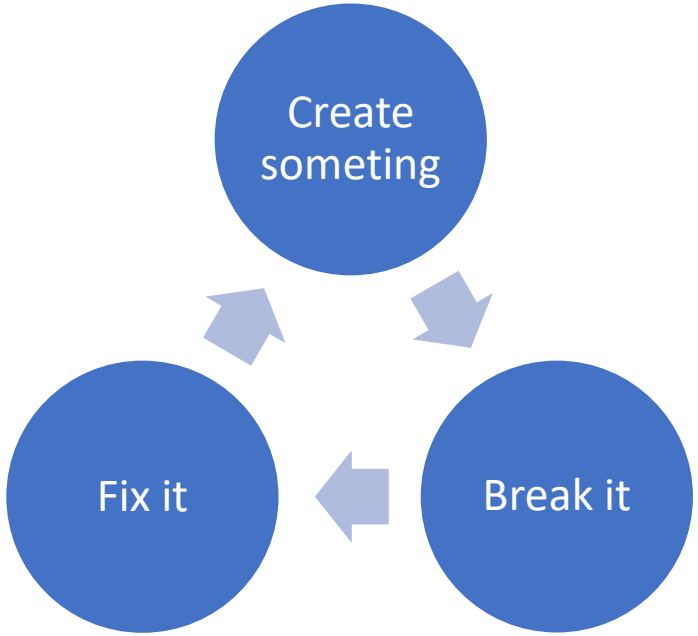
Have I seen that guy before?

CI step 1: Commiting code

MLOps

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

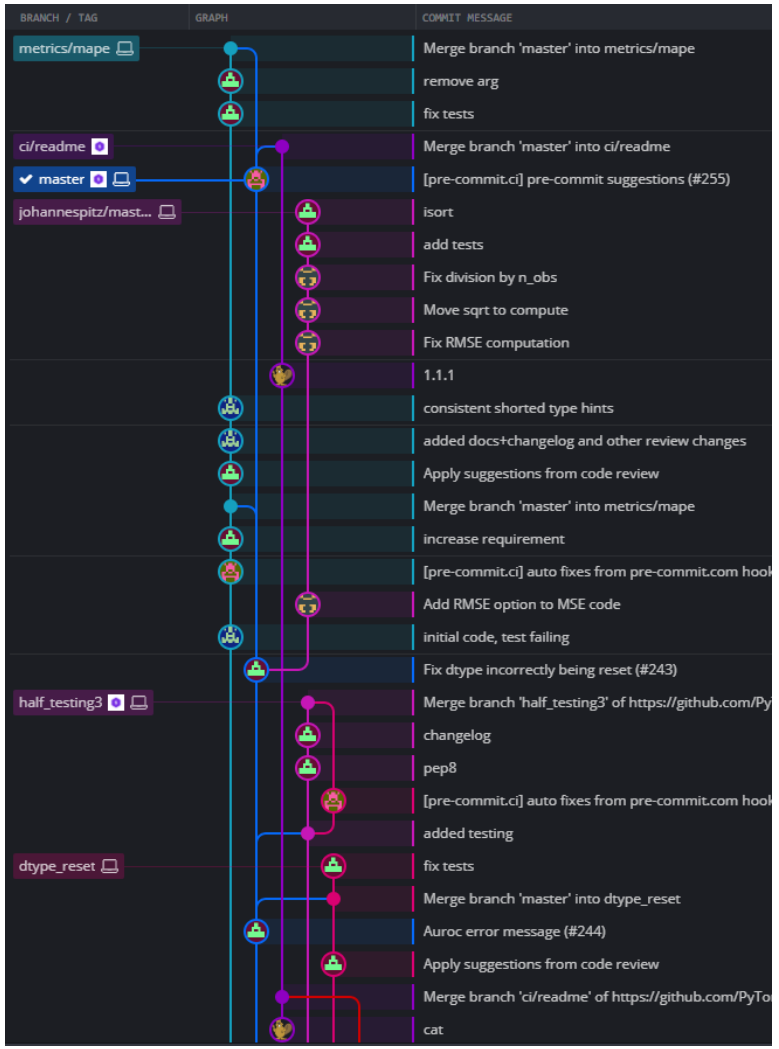




CI step 1: Commiting code

MLOps

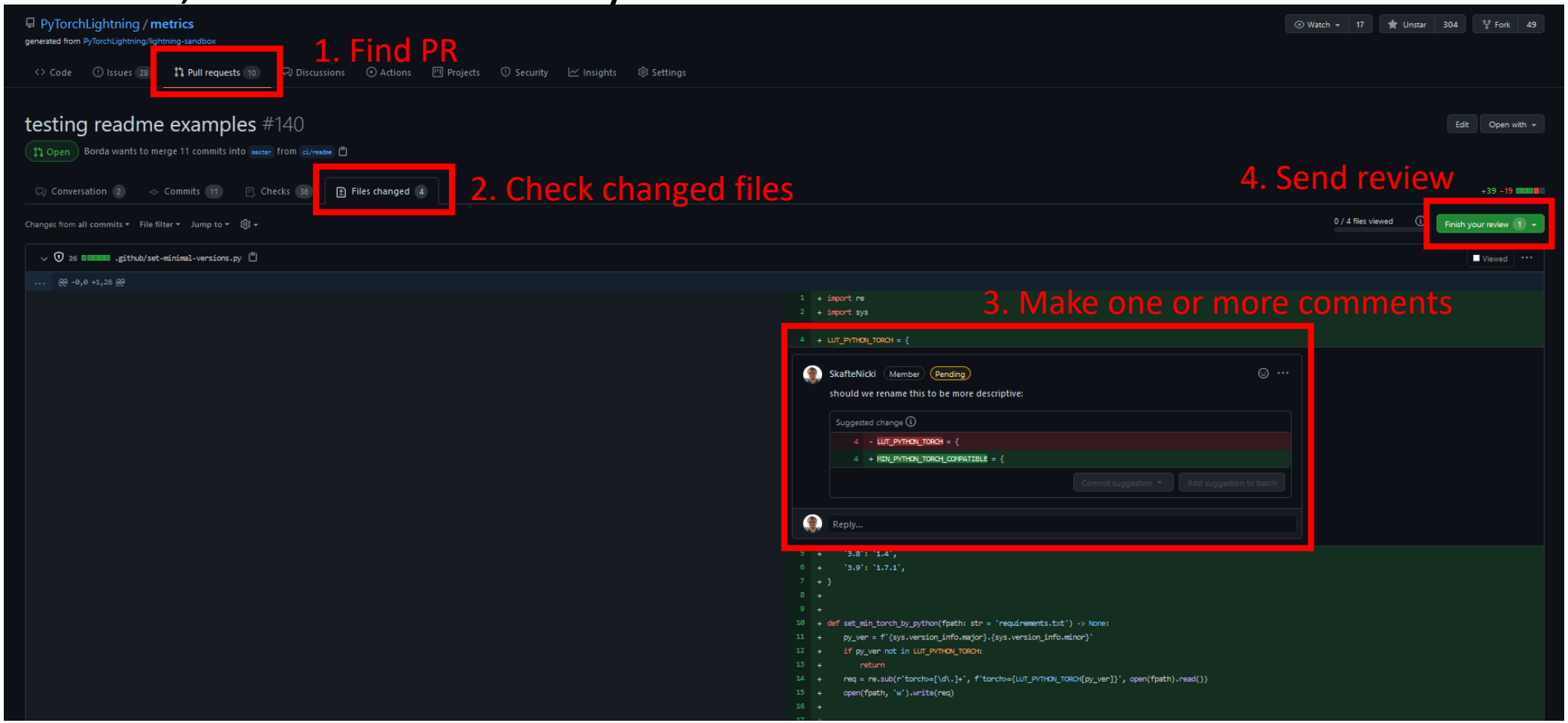
- Use branches
  - Enables parallel workflow
  - Experimental features/changes are keept away from ”stable” master



CI step 1: Commiting code

MLOps

- Use PRs, other can review your code



CI step 2: Automating stuff

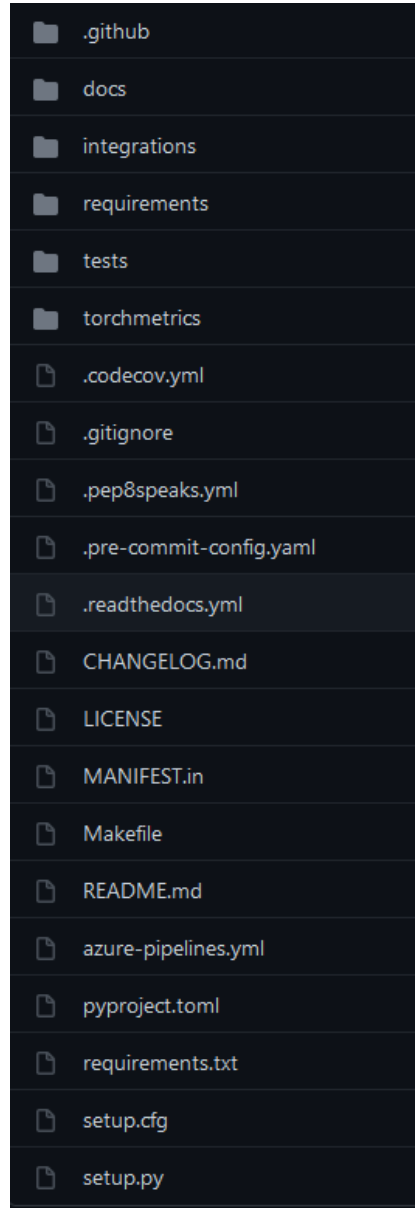
- What can be automated: EVERYTHING
  - Functional tests
  - Documentation creation
  - Linters (which check style formatting)
  - Security checks
  - Code coverage
  - Custom checks…

A small casestudy

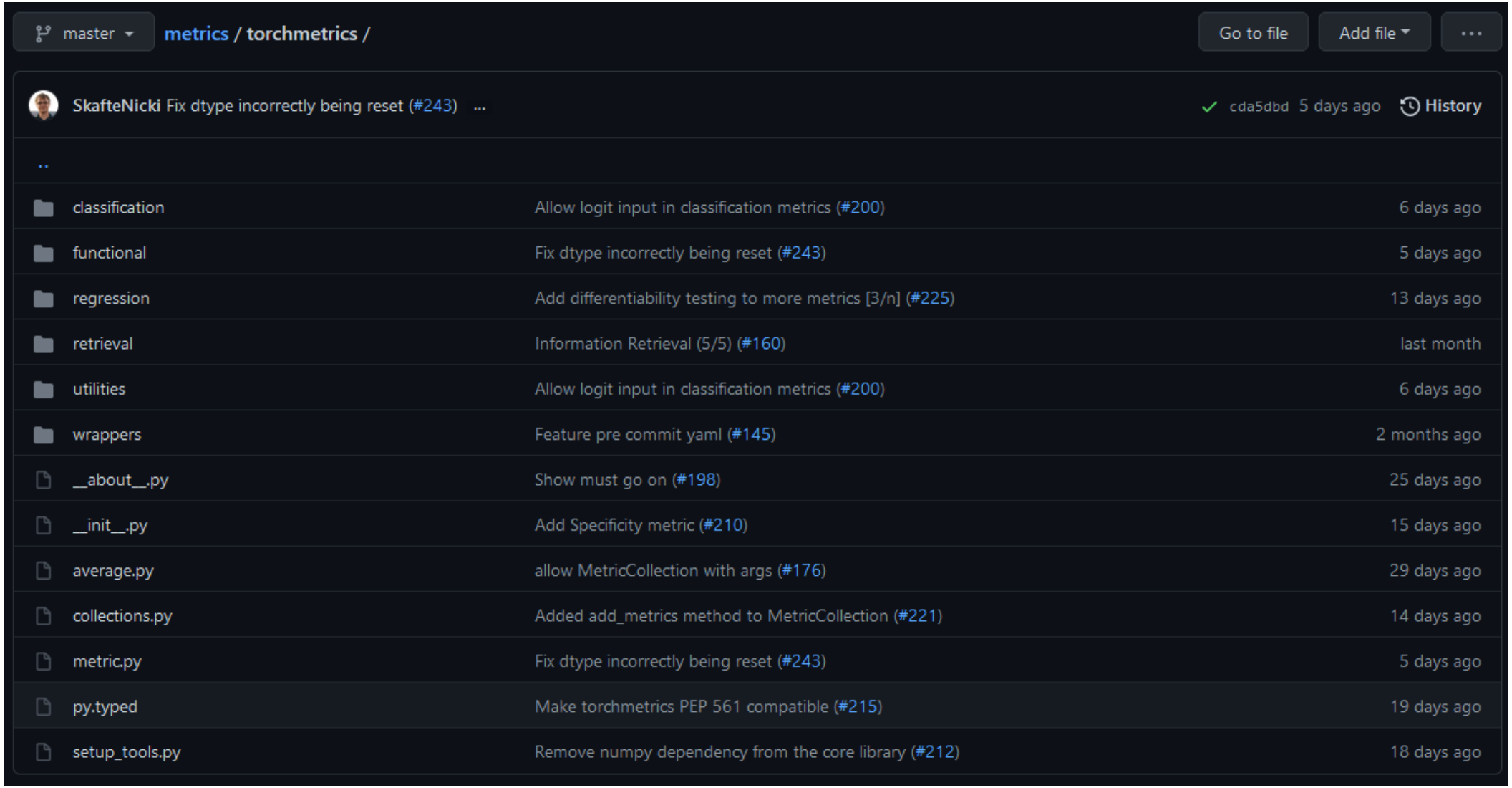
MLOps

Ranked by importance (biased)

1. Source code
2. Tests
3. Setup
4. CI workflows
5. Documentation
6. ★



Source code



master metrics / torchmetrics / Go to file Add file ...

SkafteNicki Fix dtype incorrectly being reset (#243) ... cda5dbd 5 days ago History

..

classification	Allow logit input in classification metrics (#200)	6 days ago
functional	Fix dtype incorrectly being reset (#243)	5 days ago
regression	Add differentiability testing to more metrics [3/n] (#225)	13 days ago
retrieval	Information Retrieval (5/5) (#160)	last month
utilities	Allow logit input in classification metrics (#200)	6 days ago
wrappers	Feature pre commit yaml (#145)	2 months ago
__about__.py	Show must go on (#198)	25 days ago
__init__.py	Add Specificity metric (#210)	15 days ago
average.py	allow MetricCollection with args (#176)	29 days ago
collections.py	Added add_metrics method to MetricCollection (#221)	14 days ago
metric.py	Fix dtype incorrectly being reset (#243)	5 days ago
py.typed	Make torchmetrics PEP 561 compatible (#215)	19 days ago
setup_tools.py	Remove numpy dependency from the core library (#212)	18 days ago

Test code

- In total we have 6365 tests

Test example 1

MLOps

Can be simple

```python
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')
```

Test example 2

Can be very complex

```python
@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
            }
        )
```

Setup files

```
setup(
    name='torchmetrics',
    version=about.__version__,
    description=about.__docs__,
    author=about.__author__,
    author_email=about.__author_email__,
    url=about.__homepage__,
    download_url=os.path.join(about.__homepage__, 'archive', 'master.zip'),
    license=about.__license__,
    packages=find_packages(exclude=['tests', 'docs']),
    long_description=long_description,
    long_description_content_type='text/markdown',
    include_package_data=True,
    zip_safe=False,
    keywords=['deep learning', 'machine learning', 'pytorch', 'metrics', 'AI'],
    python_requires='>=3.6',
    setup_requires=[],
    install_requires=setup_tools._load_requirements(_PATH_ROOT),
    project_urls={
        "Bug Tracker": os.path.join(about.__homepage__, 'issues'),
        "Documentation": "https://torchmetrics.rtfd.io/en/latest/",
        "Source Code": about.__homepage__,
    },
    classifiers=[
        'Environment :: Console',
        'Natural Language :: English',
        # How mature is this project? Common values are
        #   3 - Alpha, 4 - Beta, 5 - Production/Stable
        'Development Status :: 3 - Alpha',
        # Indicate who your project is intended for
        'Intended Audience :: Developers',
        'Topic :: Scientific/Engineering :: Artificial Intelligence',
        'Topic :: Scientific/Engineering :: Image Recognition',
        'Topic :: Scientific/Engineering :: Information Analysis',
        # Pick your license as you wish
        # 'License :: OSI Approved :: BSD License',
        'Operating System :: OS Independent',
        # Specify the Python versions you support here. In particular, ensure
        # that you indicate whether you support Python 2, Python 3 or both.
        'Programming Language :: Python :: 3',
        'Programming Language :: Python :: 3.6',
        'Programming Language :: Python :: 3.7',
        'Programming Language :: Python :: 3.8',
        'Programming Language :: Python :: 3.9',
    ],
)
```

- Contains all information regarding the project

- Allow people to do:

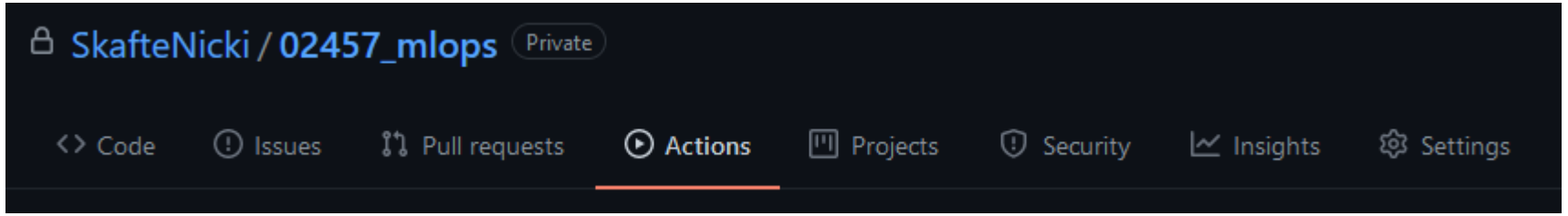
  python setup.py install

  or if uploaded to pip

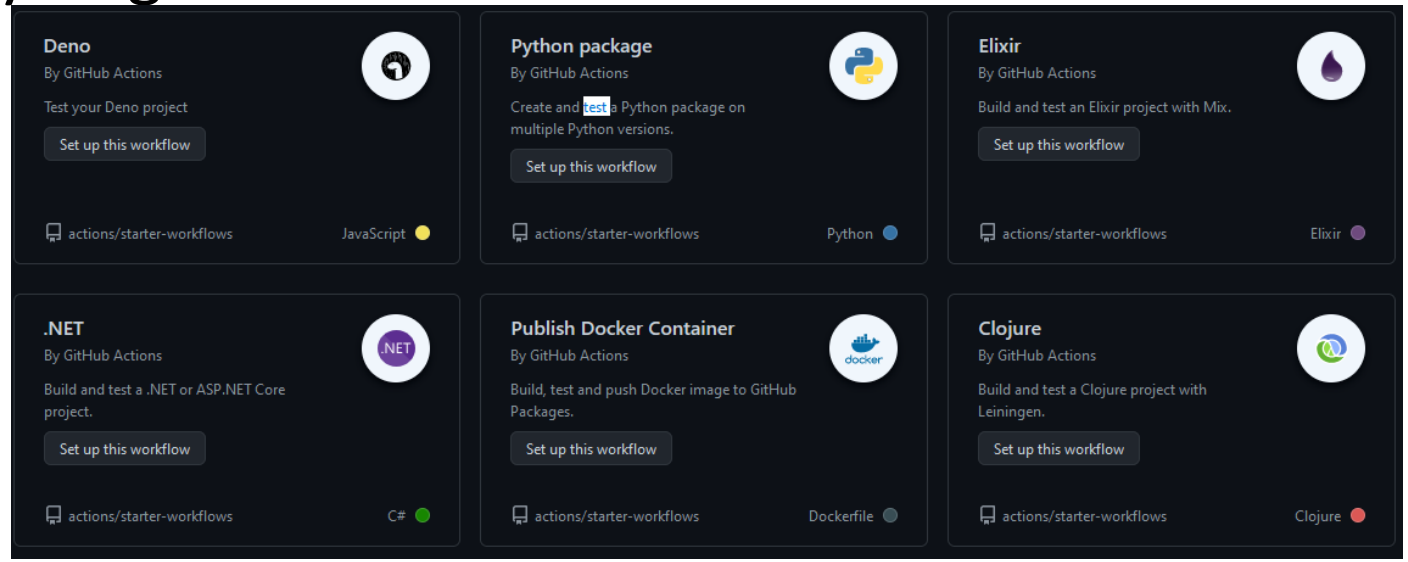
  pip install my_package

Github actions

- Build-in CI for github
- Free 2,000 automation minutes/month (public reposatory)



- Many ready to go workflows



The anatomy of a workflow file

When tests should be triggered

Define Operating system + python version

Setup python

Install dependencies

Check linting (stop if errors)

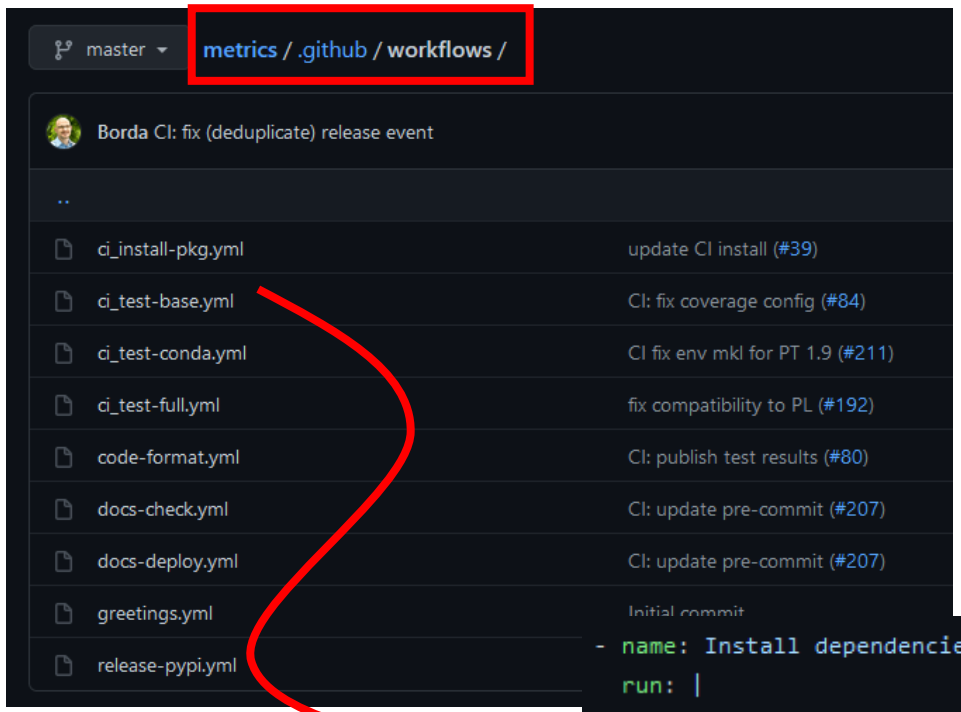
Run tests

02457_mlops / .github / workflows / python-package.yml in main

Edit new file | Preview

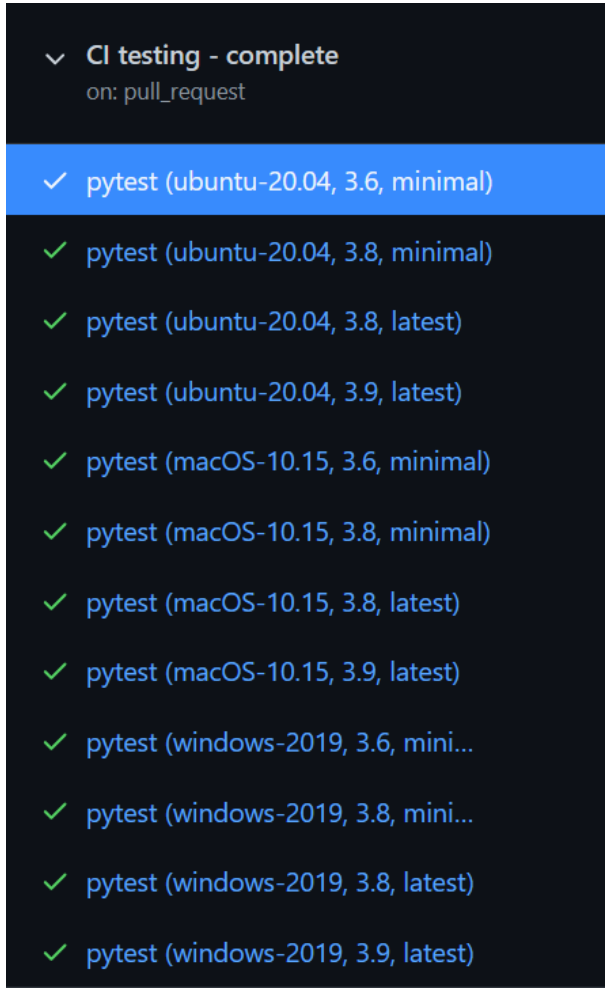
```yaml
name: Python package
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      fail-fast: false
      matrix:
        python-version: [3.7, 3.8, 3.9]
    steps:
    - uses: actions/checkout@v2
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        python -m pip install flake8 pytest
        if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
    - name: Lint with flake8
      run: |
        # stop the build if there are Python syntax errors or undefined names
        flake8 . --count --select=E9,F63,F7,F82 --show-source --statistics
        # exit-zero treats all errors as warnings. The GitHub editor is 127 chars wide
        flake8 . --count --exit-zero --max-complexity=10 --max-line-length=127 --statistics
    - name: Test with pytest
      run: |
        pytest
```

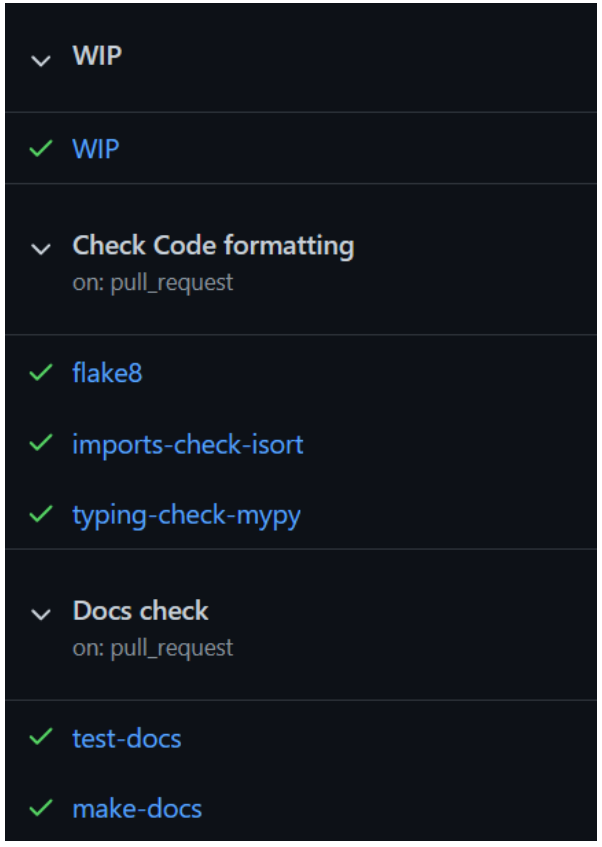
A small case study

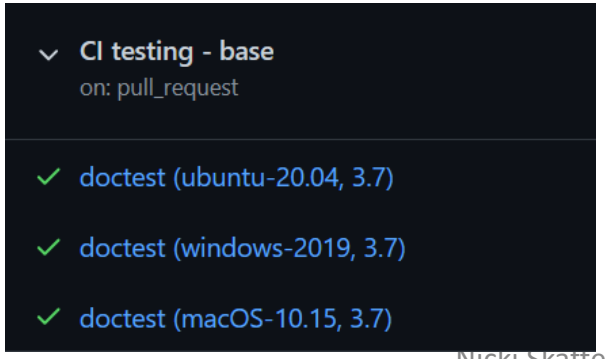


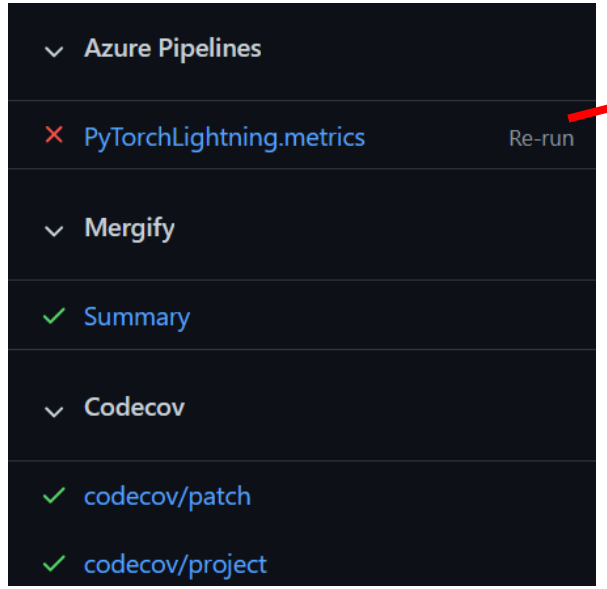

```yaml
- name: Install dependencies
  run: |
    python -m pip install --upgrade --user pip
    pip install --requirement ./requirements.txt --find-links https://download.pytorch.org/whl/cpu/torch_stable.html
    pip install "pytest>6.0" "pytest-cov>2.10" --upgrade-strategy only-if-needed
    python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

- name: Test Package [only]
  run: |
    # NOTE: run coverage on tests does not propagare 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
```

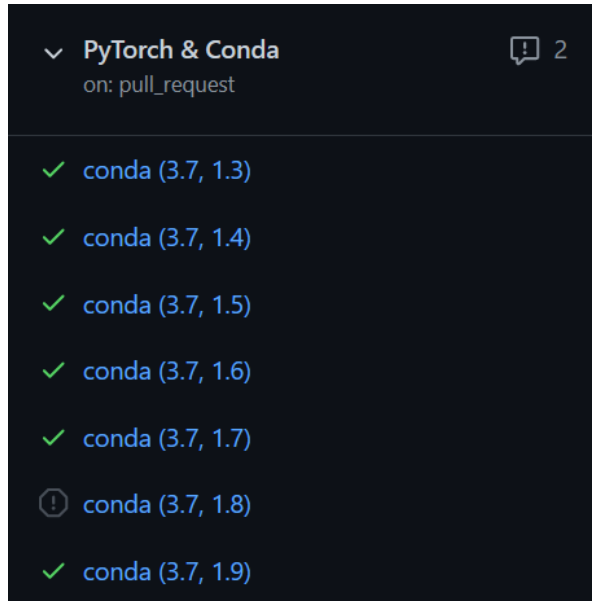
All of our workflows – 36 in total



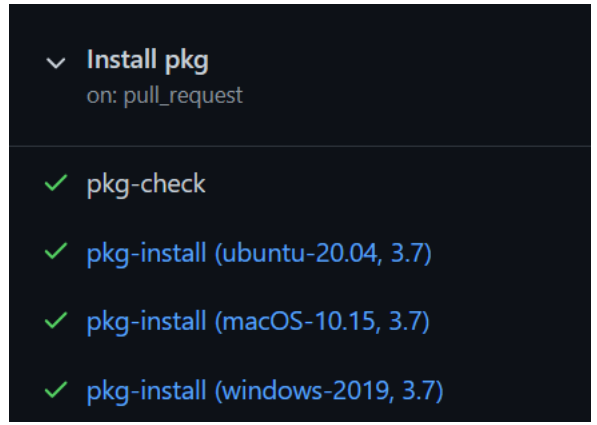








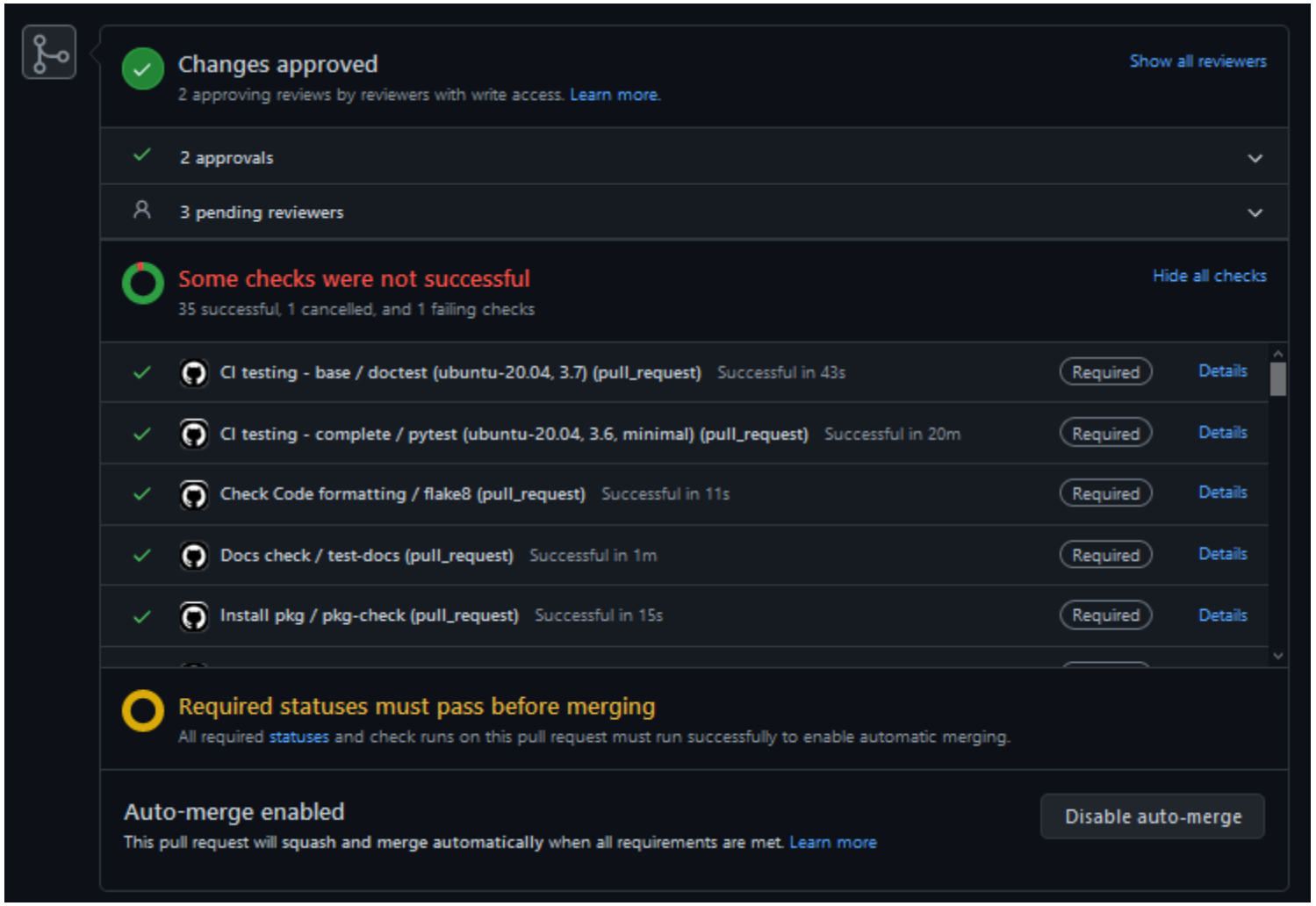
Not github actions



Test combination of
- Hardware setup
- Operating system
- Python version
- Dependencies

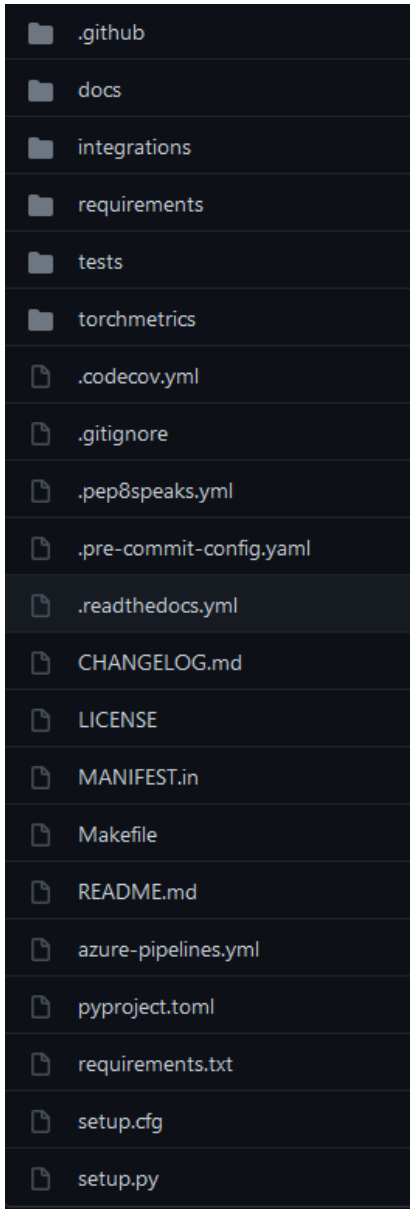
Runs tests, docs, coverage, lintint, package install ect

Each PR triggers all tests



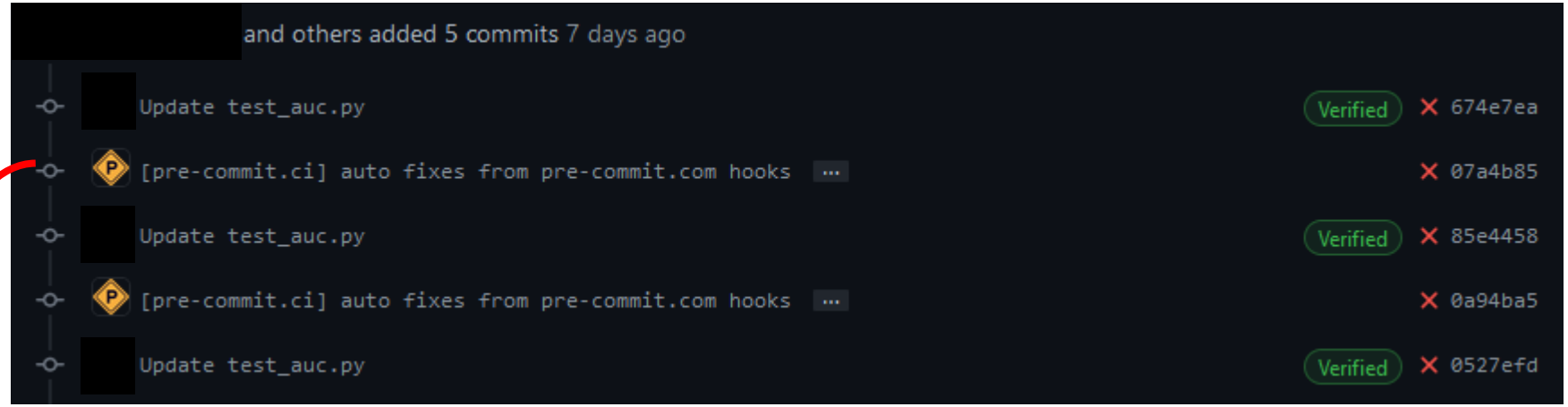
Other files to consider

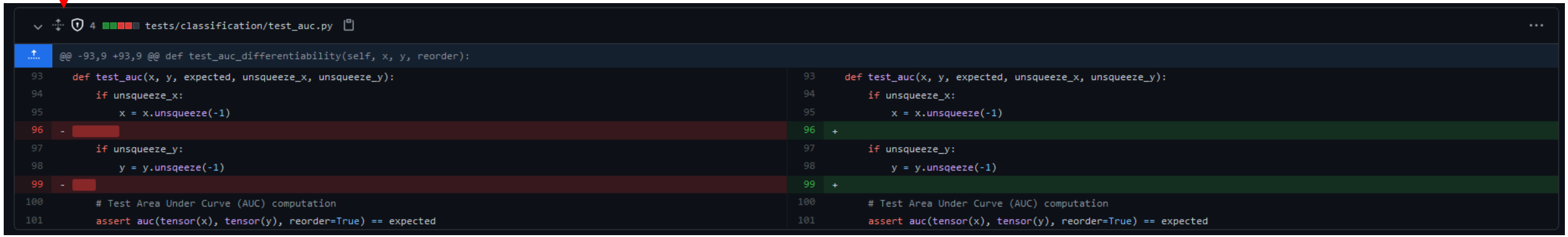
- reguirements.txt
- LICENSE
- README.md
- .gitignore



Advanced: Use bots

Bots can take care of tidious task for you (like linting)





The future is here

MLOps