

Lab 0: Fundamentals of Continuous Integration

Master's in Machine Learning 2025

Author: Igor Vons

Date: November 14, 2025

1 Test Implementation

The strategy followed to implement the tests involved creating a suite of unit tests using the pytest framework. Each test in tests/test_logic.py targets a specific function from the preprocessing module. Most are simple input-output tests, ensuring that given a specific input, the function produces the expected output.

Likewise, the CLI commands were tested using click's CliRunner, which simulates command-line invocations and captures their output for verification. For each test present in tests/test_logic.py there is a corresponding test in tests/test_cli.py that verifies the CLI command's behavior.

Some of the outputs of the cli are expected to differ from the bare logic function outputs due to formatting, as in command-line outputs are strings, while logic function outputs can be lists or other data structures. The assert statements in the tests take this into account.

2 Test outputs

The tests were executed using the `uv run python -m pytest -v` command in the terminal. All tests passed successfully, indicating that both the logic functions and CLI commands are functioning as intended. Below we can see the output of the test execution:

```
(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pytest -v
===== test session starts =====
platform linux -- Python 3.13.7, pytest-8.4.2, pluggy-1.6.0 -- /home/ivons/molopes/.venv/bin/python3
cachedir: .pytest_cache
rootdir: /home/ivons/molopes
configfile: pyproject.toml
plugins: cov-7.0.0
collected 35 items

test/test_cli.py::test_cli_clean_command[cleaning clean-input_data0-Cleaned Data: ['1', '2', 'None', '4', '5', 'nan', '6']\n] PASSED [ 2%]
test/test_cli.py::test_cli_clean_command[cleaning clean-input_data1-Cleaned Data: ['a', 'b', 'None', 'c']\n] PASSED [ 5%]
test/test_cli.py::test_cli_fill_command_custom[cleaning fill-input_data0-options0-Filled Data: ['1', '2', 'None', '4', -1, '5', 'nan', '6']\n] PASSED [ 8%]
test/test_cli.py::test_cli_fill_command_custom[cleaning fill-input_data1-options1-Filled Data: ['a', 'x', 'b', 'None', 'c']\n] PASSED [ 11%]
test/test_cli.py::test_cli_fill_command_default[cleaning fill-input_data0-Filled Data: ['1', '2', 'None', '4', 0, '5', 'nan', '6']\n] PASSED [ 14%]
test/test_cli.py::test_cli_fill_command_default[cleaning fill-input_data1-Filled Data: ['a', 0, 'b', 'None', 'c']\n] PASSED [ 17%]
test/test_cli.py::test_cli_minmax_normalize_command PASSED [ 20%]
test/test_cli.py::test_cli_z_score_standardize_command PASSED [ 22%]
test/test_cli.py::test_cli_clip_values_command PASSED [ 25%]
test/test_cli.py::test_cli_lst_to_ints_command PASSED [ 28%]
test/test_cli.py::test_cli_log_transform_command PASSED [ 31%]
test/test_cli.py::test_cli_special_tokenization_command PASSED [ 34%]
test/test_cli.py::test_cli_clean_text_command PASSED [ 37%]
test/test_cli.py::test_cli_clean_stop_words_command PASSED [ 40%]
test/test_cli.py::test_cli_lst_shuffle_command PASSED [ 42%]
test/test_cli.py::test_cli_lst_flatten_command PASSED [ 45%]
test/test_cli.py::test_cli_remove_duplicates_command PASSED [ 48%]
test/test_cli.py::test_cli_invalid_command PASSED [ 51%]
test/test_logic.py::test_clean_values[input_data0-expected0] PASSED [ 54%]
test/test_logic.py::test_clean_values[input_data1-expected1] PASSED [ 57%]
test/test_logic.py::test_fill_values_custom[input_data0--1-expected0] PASSED [ 60%]
test/test_logic.py::test_fill_values_custom[input_data1-x-expected1] PASSED [ 62%]
test/test_logic.py::test_fill_values_default[input_data0-expected0] PASSED [ 65%]
test/test_logic.py::test_fill_values_default[input_data1-expected1] PASSED [ 68%]
test/test_logic.py::test_minmax_normalize PASSED [ 71%]
test/test_logic.py::test_z_score_standardize PASSED [ 74%]
test/test_logic.py::test_clip_values PASSED [ 77%]
test/test_logic.py::test_lst_to_ints PASSED [ 80%]
test/test_logic.py::test_log_transform PASSED [ 82%]
test/test_logic.py::test_special_tokenization PASSED [ 85%]
test/test_logic.py::test_clean_text PASSED [ 88%]
test/test_logic.py::test_clean_stop_words PASSED [ 91%]
test/test_logic.py::test_lst_shuffle PASSED [ 94%]
test/test_logic.py::test_lst_flatten PASSED [ 97%]
test/test_logic.py::test_remove_duplicates PASSED [100%]

===== 35 passed in 0.13s =====
```

Figure 1: Output of test execution showing all tests passing successfully.

3 Test coverage

To check the test coverage, the `pytest-cov` plugin was used. The command `uv run python -m pytest -v --cov=src` gives us some extra info on top of the test results:

```
===== tests coverage =====
coverage: platform linux, python 3.13.7-final-0

Name                Stmts   Miss  Cover
-----
src/__init__.py         2      0   100%
src/cli.py             100     0   100%
src/preprocessing.py    71      5    93%
TOTAL                  173      5    97%

35 passed in 0.35s
```

Figure 2: Output of test coverage analysis showing 100% coverage across all modules.

We can see although most of the preprocessing module is covered, we are still missing a few (5 out of 71). This should be addressed in future iterations to ensure complete test coverage.

4 Linter & Formatter

The codebase was linted and formatted using `black` for formatting and `pylint` for linting. The commands used were `uv run python -m pylint src/*.py` and `uv run black src/*.py`. Both tools reported no issues, indicating that the code adheres to the specified style guidelines and is free of common coding errors.

```
(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pylint src/*.py
***** Module src.__main__
src/__main__.py:5:0: C0304: Final newline missing (missing-final-newline)
***** Module src.cli
src/cli.py:47:0: C0301: Line too long (108/100) (line-too-long)
src/cli.py:54:0: C0301: Line too long (110/100) (line-too-long)
src/cli.py:51:7: C0123: Use isinstance() rather than type() for a typecheck. (unidiomatic-typecheck)
src/cli.py:53:7: C0123: Use isinstance() rather than type() for a typecheck. (unidiomatic-typecheck)
***** Module src.preprocessing
src/preprocessing.py:69:0: C0303: Trailing whitespace (trailing-whitespace)
src/preprocessing.py:33:12: R1701: Consider merging these isinstance calls to isinstance(item, (float, int)) (consider-merging-isinstance)
src/preprocessing.py:65:14: R1701: Consider merging these isinstance calls to isinstance(item, (float, int)) (consider-merging-isinstance)

-----
Your code has been rated at 9.42/10 (previous run: 10.00/10, -0.58)

(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pylint src/*.py
***** Module src.__main__
src/__main__.py:5:0: C0304: Final newline missing (missing-final-newline)
***** Module src.preprocessing
src/preprocessing.py:68:36: C0303: Trailing whitespace (trailing-whitespace)

-----
Your code has been rated at 9.86/10 (previous run: 9.42/10, +0.43)

(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pylint src/*.py
***** Module src.preprocessing
src/preprocessing.py:68:36: C0303: Trailing whitespace (trailing-whitespace)

-----
Your code has been rated at 9.93/10 (previous run: 9.86/10, +0.07)

(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pylint src/*.py
-----
Your code has been rated at 10.00/10 (previous run: 9.93/10, +0.07)
```

Figure 3: Output of pylint, displaying the last iterations until no more issues were found.

```
(mlopslab) ivons@igorvons:~/molopes$ uv run black src/*.py
reformatted src/__init__.py
reformatted src/preprocessing.py
reformatted src/cli.py

All done! ✨ 🍰 ✨
3 files reformatted.
(mlopslab) ivons@igorvons:~/molopes$ uv run python -m pylint src/*.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Figure 4: Output of black, showing that linter reported no formatting issues after running.