

Issue #22313: `check_estimator` running tests which shouldn't run when `non_deterministic: True`

Link to Issue: <https://github.com/scikit-learn/scikit-learn/issues/22313>

Summary of issue: Performing `pytest` on a third party estimators, (i.e an estimator that is NOT in `scikit-learn` and is built by the user) with tag `non_deterministic` set to `True`, should NOT run two of the following tests (`check_methods_sample_order_invariance()` and `check_methods_subset_invariance()`).

*Note: This issue was fixed on `Scikit-learn`'s official github during the time that we worked on this issue and when we cloned the `Scikit-learn` repo onto our D01 repo, therefore in our repo the edits on this issue might seem weird but the above is the reason.

Test code & methodology: See <https://github.com/scikit-learn/scikit-learn/issues/22313>

1. Create a new python file, containing the test code, and save it in the build directory.
Notice the `non_deterministic` tag is set to `True`.
2. Run the “`check_methods_sample_order_invariance`” and “`check_methods_subset_invariance`” test cases with the `pytest` framework. This can be done with “`pytest [file_name] -k check_methods_sample_order_invariance`” and “`pytest [file_name] -k check_methods_subset_invariance`”
3. Repeat the same step with the `non_deterministic` tag set to `False`.

Reproducing the bug

Running the code above with `pytest` on the current build of `scikit learn` will result in 2 failed test cases (See image below).

```

E       Not equal to tolerance rtol=1e-07, atol=1e-07
E       predict of TemplateEstimator is not invariant when applied to a subset.
E       Mismatched elements: 12 / 20 (60%)
E       Max absolute difference: 2
E       Max relative difference: 1.
E       x: array([1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 2, 1, 2, 0, 2, 1, 2, 0, 0])
E       y: array([2, 0, 1, 2, 2, 0, 1, 1, 2, 0, 2, 1, 1, 1, 1, 2, 1, 1, 0, 0])

scikit-learn/scikit-learn/sklearn/utils/estimator_checks.py:1316: AssertionError
===== warnings summary =====
a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
/mnt/c/Users/Kelvin/Desktop/School/CSCD01/scikit-learn/sklearn-env/lib/python3.8/site-packages/joblib/backports.py:36:
DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if LooseVersion(np.__version__) < '1.13':

a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
/mnt/c/Users/Kelvin/Desktop/School/CSCD01/scikit-learn/sklearn-env/lib/python3.8/site-packages/setuptools/_distutils/version.py:351: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  other = LooseVersion(other)

-- Docs: https://docs.pytest.org/en/stable/warnings.html
===== short test summary info =====
FAILED a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_methods_sample_order_invariance] - Assertion...
FAILED a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_methods_subset_invariance] - AssertionError:
===== 2 failed, 28 passed, 2 skipped, 4 warnings in 10.92s =====
(sklearn-env) tankelvi@LAPTOP-MTN6NPJR:/mnt/c/Users/Kelvin/Desktop/School/CSCD01$

```

Technical details and design: In the scikit-learn community, pytest is commonly used as a testing framework. To understand the cause of the issue, we need to understand Pytest. Pytest has “parameterized fixtures”, which take the form of a decorator `pytest.mark.parametrize`.

For example (from <https://docs.pytest.org/en/6.2.x/parametrize.html>):

```
import pytest
```

```
@pytest.mark.parametrize("test_input,expected", [("3+5", 8), ("2+4", 6), ("6*9", 42)])
```

```
def test_eval(test_input, expected):
```

```
    assert eval(test_input) == expected
```

In this simple example, the `pytest.mark.parametrize` decorator takes two parameters: a string, which concatenates the variable names given to the parameters of the `test_eval` function, and a list of tuples. Notice each tuple in the list contains precisely 2 elements because `test_eval` takes in 2 parameters. The `test_eval` function will be called 3 times because there are 3 tuples. The parameter `test_input` will be filled with the first element of the tuple and `expected` will be filled with the second element from the same tuple as the first element.

The test function `parametrize_with_checks(estimators)`, located in “`sklearn/utils/estimator_checks.py`”, takes in a list of estimators and returns a

pytest.mark.parametrize decorator. For each estimator in the list, it will run `_yield_all_checks(estimator)`, defined in the same file, which returns a generator. See <https://wiki.python.org/moin/Generators> for more details on Python generators. It is a one-time iterator containing all the tests that should be run for each of the estimators.

Implementation: The `check_methods_sample_order_invariance` and `check_methods_subset_invariance` functions were always being returned by the generator in `_yield_all_checks(estimator)`. To fix the issue, we simply use the tags dictionary provided by the estimator to check the value of the `non_deterministic` tag. When the `non_deterministic` is set to True, both `check_methods_sample_order_invariance` and `check_methods_subset_invariance` are skipped and thus not included in the generators. See code changes below.

```
sklearn > utils > estimator_checks.py > ...
317 317     for check in _yield_outliers_checks(estimator):
318 318         yield check
319 319     yield check_parameters_default_constructible
320     yield check_methods_sample_order_invariance
321     yield check_methods_subset_invariance
322 320     yield check_fit2d_1sample
323 321     yield check_fit2d_1feature
324 322     yield check_get_params_invariance

320     yield check_fit2d_1sample
321     yield check_fit2d_1feature
322     yield check_get_params_invariance
323     yield check_set_params
324     yield check_dict_unchanged
325     yield check_dont_overwrite_parameters
326     yield check_fit_idempotent
327     yield check_fit_check_is_fitted
328     if not tags["no_validation"]:
329         yield check_n_features_in
330         yield check_fit1d
331         yield check_fit2d_predict1d
332         if tags["requires_y"]:
333             yield check_requires_y_none
334         if tags["requires_positive_X"]:
335             yield check_fit_non_negative
336     # When the non_deterministic tag is set to true,
337     # ignore these two test cases
338     if not tags["non_deterministic"]:
339         yield check_methods_sample_order_invariance
340         yield check_methods_subset_invariance
```

Verification

After rebuilding and running the same test code (`non_deterministic` set to **True**), the issue is now fixed and working as expected.

```
TERMINAL  PROBLEMS 11  OUTPUT  DEBUG CONSOLE

if LooseVersion(np.__version__) < '1.13':

a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
a.py::test_sklearn_compatible_estimator[TemplateEstimator()-check_estimators_fit_returns_self(readonly_memmap=True)]
/mnt/c/Users/Kelvin/Desktop/School/CSCD01/scikit-learn/sklearn-env/lib/python3.8/site-packages/setuptools/_distutils/version.py:351: DeprecationWarning: distuti
ls Version classes are deprecated. Use packaging.version instead.
other = LooseVersion(other)

-- Docs: https://docs.pytest.org/en/stable/warnings.html
===== 28 passed, 2 skipped, 4 warnings in 9.83s =====
(sklearn-env) tankelv1@LAPTOP-MTNGNPJR: /mnt/c/Users/Kelvin/Desktop/School/CSCD01$
```

Once again, check that `check_methods_sample_order_invariance()` and `check_methods_subset_invariance()` runs and passes on estimators with `non_deterministic` set to False. Check with LinearRegression (non-third party estimator).

```
(sklearn-env) tankelv1@LAPTOP-MTNGNPJR: /mnt/c/Users/Kelvin/Desktop/School/CSCD01$ cat b.py
from sklearn.linear_model import LinearRegression
from sklearn.utils.estimator_checks import parametrize_with_checks

@parametrize_with_checks([LinearRegression()])
def test_sklearn_compatible_estimator(estimator, check):
    check(estimator)

(sklearn-env) tankelv1@LAPTOP-MTNGNPJR: /mnt/c/Users/Kelvin/Desktop/School/CSCD01$ pytest b.py -k check_methods_sample_order_invariance
===== test session starts =====
platform linux -- Python 3.8.0, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: /mnt/c/Users/Kelvin/Desktop/School/CSCD01
collected 48 items / 47 deselected / 1 selected

b.py . [100%]

===== 1 passed, 47 deselected in 7.21s =====
(sklearn-env) tankelv1@LAPTOP-MTNGNPJR: /mnt/c/Users/Kelvin/Desktop/School/CSCD01$ pytest b.py -k check_methods_subset_invariance
===== test session starts =====
platform linux -- Python 3.8.0, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: /mnt/c/Users/Kelvin/Desktop/School/CSCD01
collected 48 items / 47 deselected / 1 selected

b.py . [100%]

===== 1 passed, 47 deselected in 8.01s =====
(sklearn-env) tankelv1@LAPTOP-MTNGNPJR: /mnt/c/Users/Kelvin/Desktop/School/CSCD01$
```