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()*).

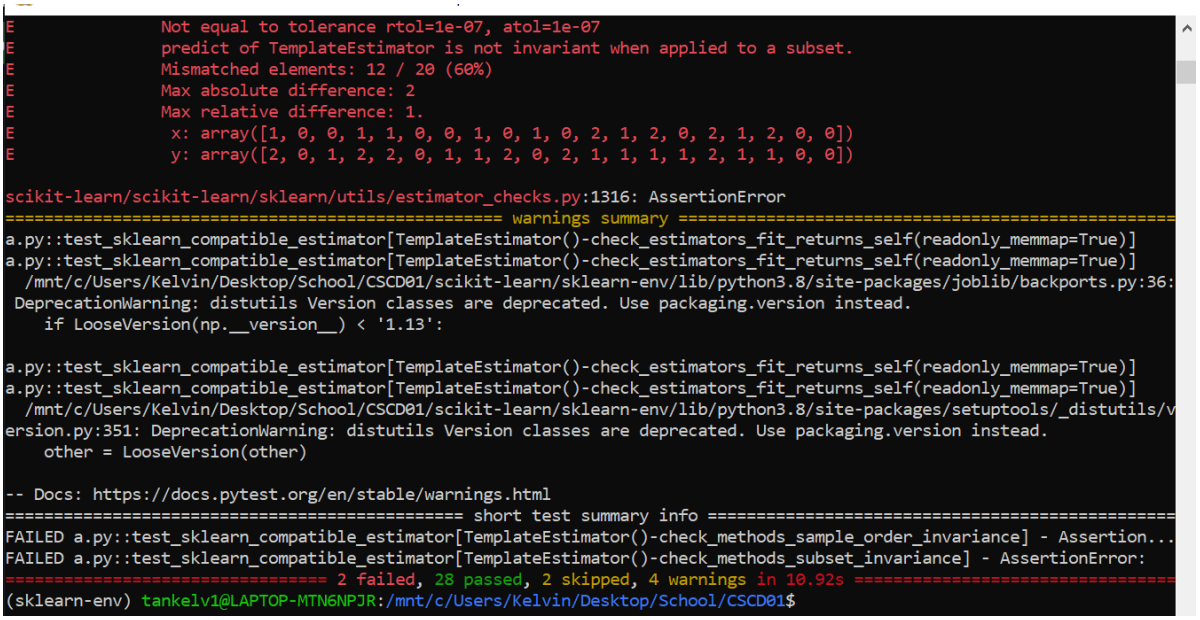
*Note: This issue was fixed on Scikit-learn's official github during the time that we worked on this issue and when we cloned the Scikit-learn repo onto our D01 repo, therefore in our repo the edits on this issue might seem weird but the above is the reason.

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).



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.

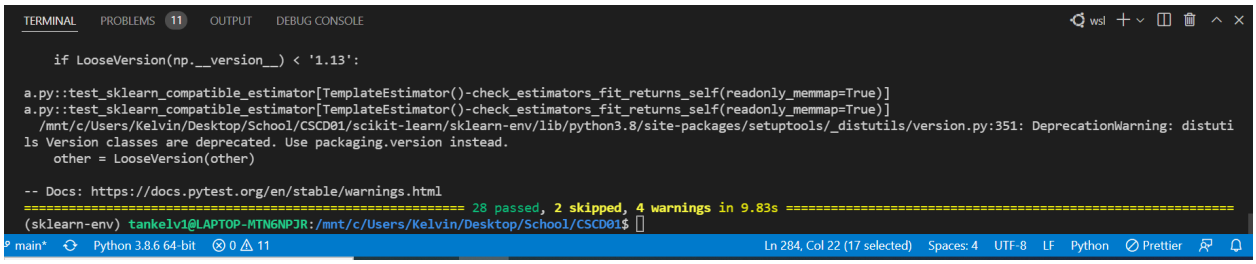
The test function *parametrize_with_checks(estimators)*, located in "sklearn/utils/estimator_checks.py", takes in a list of estimators and returns a

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.

Verification

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



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).