Issue #16710: Pipeline requires both fit and transform method to be available instead of only fit_transform

Link to Issue: https://github.com/scikit-learn/scikit-learn/issues/16710

Summary of issue: Calling a pipeline with a nonparametric function raises an error since the function transform() is missing. The pipeline itself calls the function fit_transform() if it's present. For nonparametric functions (the most prominent being t-SNE) a regular transform() method does not exist since there is no projection or mapping that is learned. But it could still be used for dimensionality reduction.

Testing Environment

// Assuming user is running python 3.9+ and sklearn 1.0+ in a similar environment

```
>>> import sklearn; sklearn.show_versions()
System:
    python: 3.9.10 | packaged by conda-forge | (main, Feb  1 2022, 21:21:54) [MSC v.1929 64 bit (AMD64)]
executable:/sklearn-env/python.exe
   machine: Windows-10-10.0.19044-SP0

Python dependencies:
          pip: 22.0.3
   setuptools: 60.9.3
      sklearn: 1.1.dev0
        numpy: 1.22.2
        scipy: 1.8.0
       Cython: 0.29.28
       pandas: None
   matplotlib: None
       joblib: 1.1.0
threadpoolctl: 3.1.0
      pytest: 7.0.1

Built with OpenMP: True
```

Resources

```
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.pipeline import make_pipeline
import numpy as np
```

Test Case

// Input in python console

```
>>> X = np.array([[1], [2]])
>>> y = np.array([1, 2])
>>> pipe = make_pipeline(TSNE(), PCA())
>>> pipe.fit(X, y)
```

Expected Output

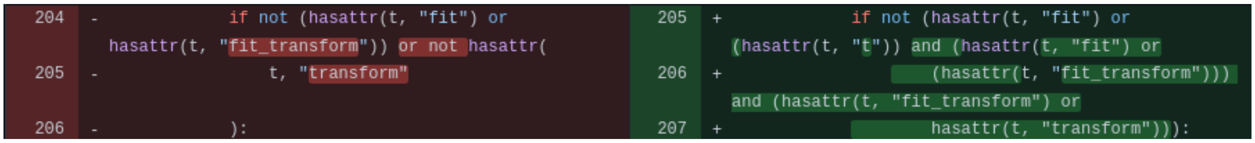
// A pipeline is created with warning message

```
…/course-project-keycap-guardians/scikit-learn/sklearn/pipeline.py:215: UserWarning: ‘TSNE()’ (type <class ‘sklearn.manifold._t_sne.TSNE’>) does not implement transform.
Pipeline(steps=[('tsne', TSNE()), ('pca', PCA())])
```

Implementation

File modified: scikit-learn/sklearn/pipeline.py

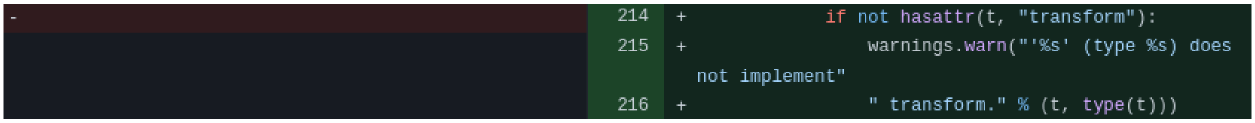
Modified logic of the code to allow pipelines with missing transformer implementation



```
-             if not (hasattr(t, "fit") or hasattr(t, "fit_transform")) or not hasattr(
-                 t, "transform"
-             ):
```

```
+             if not (hasattr(t, "fit") or (hasattr(t, "t")) and (hasattr(t, "fit") or
+                     (hasattr(t, "fit_transform"))) and (hasattr(t, "fit_transform") or
+                     hasattr(t, "transform"))):
```

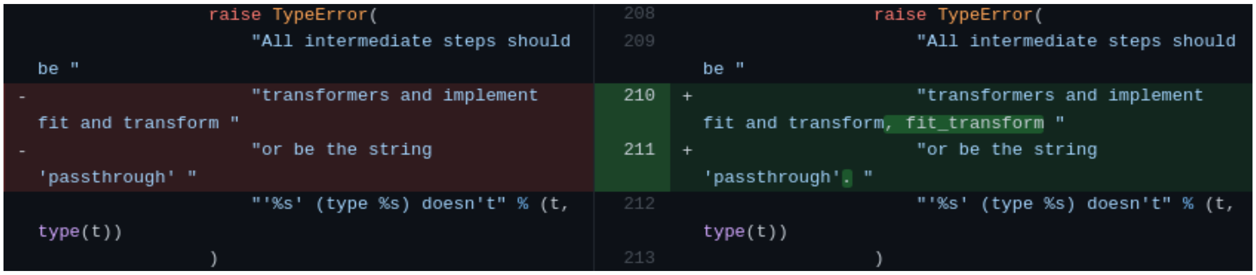
Added warnings for when transform is not implemented



```
-
```

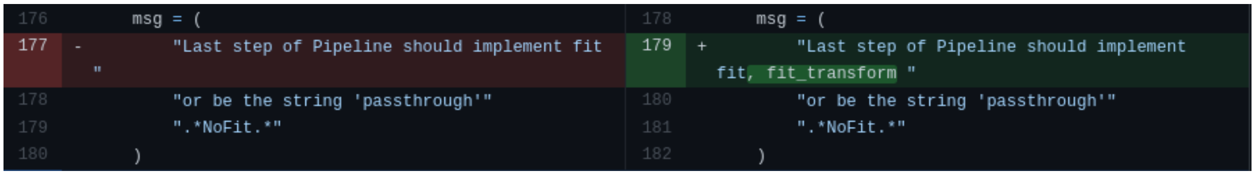
```
+             if not hasattr(t, "transform"):
+                 warnings.warn("'%s' (type %s) does not implement"
+                 " transform." % (t, type(t)))
```

Adjusted error messages to inform users that the creation of a pipeline is now allowed.



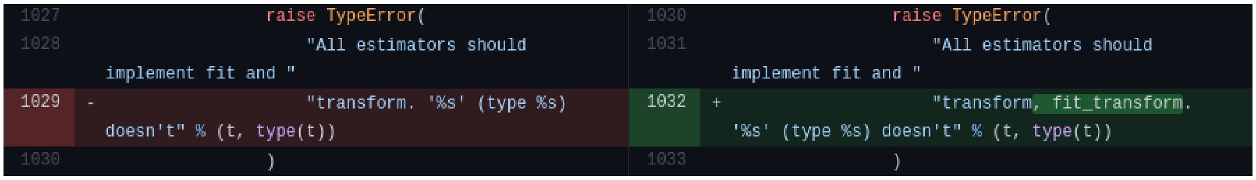
```
                  raise TypeError(
                      "All intermediate steps should be "
-                     "transformers and implement fit and transform "
-                     "or be the string 'passthrough' "
                      "'%s' (type %s) doesn't" % (t, type(t))
                  )
```

```
                  raise TypeError(
                      "All intermediate steps should be "
+                     "transformers and implement fit and transform, fit_transform "
+                     "or be the string 'passthrough'. "
                      "'%s' (type %s) doesn't" % (t, type(t))
                  )
```



```
        msg = (
-           "Last step of Pipeline should implement fit "
            "or be the string 'passthrough'"
            ".*NoFit.*"
        )
```

```
        msg = (
+           "Last step of Pipeline should implement fit, fit_transform "
            "or be the string 'passthrough'"
            ".*NoFit.*"
        )
```

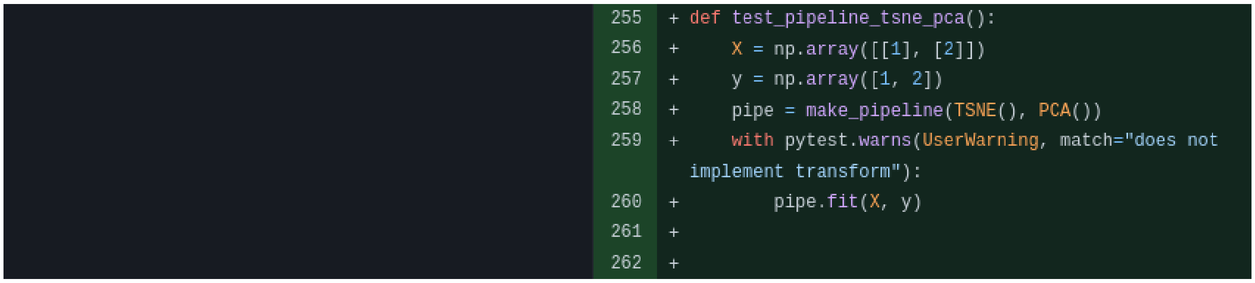


```
                raise TypeError(
                    "All estimators should implement fit and "
-                   "transform. '%s' (type %s) doesn't" % (t, type(t))
                )
```

```
                raise TypeError(
                    "All estimators should implement fit and "
+                   "transform, fit_transform. '%s' (type %s) doesn't" % (t, type(t))
                )
```

File modified: scikit-learn/sklearn/tests/test_pipeline.py

Added test to check that a warning is raised when a pipe is trying to fit when there is a missing transform implementation.



```
+ def test_pipeline_tsne_pca():
+     X = np.array([[1], [2]])
+     y = np.array([1, 2])
+     pipe = make_pipeline(TSNE(), PCA())
+     with pytest.warns(UserWarning, match="does not implement transform"):
+         pipe.fit(X, y)
+
+
```