/home/subhodeep/projects/mgproj/br/lightning/lightning/tests/te dataset.py

```
import numpy as np
import scipy.sparse as sp

from sklearn.utils.testing import assert_array_equal
from sklearn.utils.testing import assert_array_almost_equal
from sklearn.utils.testing import assert_equal

from sklearn.datasets.samples_generator import make_classification
from sklearn.utils import check_random_state

from lightning.dataset_fast import ContiguousDataset
from lightning.dataset_fast import FortranDataset
from lightning.dataset_fast import CSRDataset
from lightning.dataset_fast import CSCDataset

# Create test datasets.
X, _ = make_classification(n_samples=20, n_features=100,
                           n_informative=5, n_classes=2, random_state=0)
X2, _ = make_classification(n_samples=10, n_features=100,
                            n_informative=5, n_classes=2, random_state=0)

# Sparsify datasets.
X[X < 0.3] = 0

X_csr = sp.csr_matrix(X)
X_csc = sp.csc_matrix(X)

rs = check_random_state(0)


def test_contiguous_get_row():
    ind = np.arange(X.shape[1])
    ds = ContiguousDataset(X)
    for i in xrange(X.shape[0]):
        indices, data, n_nz = ds.get_row(i)
        assert_array_equal(indices, ind)
        assert_array_equal(data, X[i])
        assert_equal(n_nz, X.shape[1])


def test_csr_get_row():
    ds = CSRDataset(X_csr)
```

```
    for i in xrange(X.shape[0]):
        indices, data, n_nz = ds.get_row(i)
        for jj in xrange(n_nz):
            j = indices[jj]
            assert_equal(X[i, j], data[jj])


def test_fortran_get_column():
    ind = np.arange(X.shape[0])
    ds = FortranDataset(np.asfortranarray(X))
    for j in xrange(X.shape[1]):
        indices, data, n_nz = ds.get_column(j)
        assert_array_equal(indices, ind)
        assert_array_equal(data, X[:, j])
        assert_equal(n_nz, X.shape[0])


def test_csc_get_column():
    ds = CSCDataset(X_csc)
    for j in xrange(X.shape[1]):
        indices, data, n_nz = ds.get_column(j)
        for ii in xrange(n_nz):
            i = indices[ii]
            assert_equal(X[i, j], data[ii])
```