Split arrays or matrices into random train and test subsets

Quick utility that wraps input validation and next(ShuffleSplit().split(X, y)) and application to input data into a single call for splitting (and optionally subsampling) data in a oneliner.

Read more in the [User Guide](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation).

**Parameters**

**\*arrays*sequence of indexables with same length / shape[0]***

Allowed inputs are lists, numpy arrays, scipy-sparse matrices or pandas dataframes.

**test\_size*float or int, default=None***

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples. If None, the value is set to the complement of the train size. If train\_size is also None, it will be set to 0.25.

**train\_size*float or int, default=None***

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split. If int, represents the absolute number of train samples. If None, the value is automatically set to the complement of the test size.

**random\_state*int or RandomState instance, default=None***

Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls. See [Glossary](https://scikit-learn.org/stable/glossary.html#term-random-state).

**shuffle*bool, default=True***

Whether or not to shuffle the data before splitting. If shuffle=False then stratify must be None.

**stratify*array-like, default=None***

If not None, data is split in a stratified fashion, using this as the class labels.

**Returns**

**splitting*list, length=2 \* len(arrays)***

List containing train-test split of inputs.

*New in version 0.16:*If the input is sparse, the output will be ascipy.sparse.csr\_matrix. Else, output type is the same as the input type.

**Examples**

>>>

**>>> import** **numpy** **as** **np**

**>>> from** **sklearn.model\_selection** **import** train\_test\_split

**>>>** X, y = np.arange(10).reshape((5, 2)), range(5)

**>>>** X

array([[0, 1],

[2, 3],

[4, 5],

[6, 7],

[8, 9]])

**>>>** list(y)

[0, 1, 2, 3, 4]

>>>

**>>>** X\_train, X\_test, y\_train, y\_test = train\_test\_split(

**...**  X, y, test\_size=0.33, random\_state=42)

**...**

**>>>** X\_train

array([[4, 5],

[0, 1],

[6, 7]])

**>>>** y\_train

[2, 0, 3]

**>>>** X\_test

array([[2, 3],

[8, 9]])

**>>>** y\_test

[1, 4]

>>>

**>>>** train\_test\_split(y, shuffle=**False**)

[[0, 1, 2], [3, 4]]