Using Random State

train_test_split splits arrays or matrices into random train and test subsets. That means that every time we run it without specifying random_state, we will get a different result, this is expected behaviour. For example:

Run 1:

```
a, b = np.arange(10).reshape((5, 2)), range(5)
train_test_split(a, b)

[array([[6, 7],
        [8, 9],
        [2, 3]]), array([[0, 1],
        [4, 5]]), [3, 4, 1], [0, 2]]
```

Run 2:

```
train_test_split(a, b)

[array([[0, 1],
        [6, 7],
        [2, 3]]), array([[8, 9],
        [4, 5]]), [0, 3, 1], [4, 2]]
```

It changes.

On the other hand, if we use random_state=some_number, then it is guaranteed that the output of **Run 1** will be equal to the output of **Run 2**, i.e. the split will be always the same.

It doesn't matter what the actual random_state number is 42, 0, 21, ... The important thing is that every time we use 42, we will always get the same output the first time we make the split. This is useful if we want reproducible results, for example in the documentation, so that everybody can consistently see the same numbers when they run the examples.