Predictive Analytics: Regression

Using Random State

The **train_test_split** function divides arrays or matrices into random training and test subsets. By default, if the **random_state** parameter is not specified, the function will generate different random splits each time it is executed. This is the expected behavior of the function. For example:

```
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]]
```

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

In contrast, when you specify a value for random_state (e.g., **random_state = some_number**), the train_test_split function ensures that the output of the split remains consistent across multiple runs. This means that the result obtained from the first run will be identical to the result obtained from the second run and subsequent runs. The specific value chosen for random_state (e.g., 42, 0, 21, etc.) is not important as long as it is consistently used. This feature is beneficial when you desire reproducible outcomes, such as in documentation or when you want others to consistently observe the same results when running the examples.