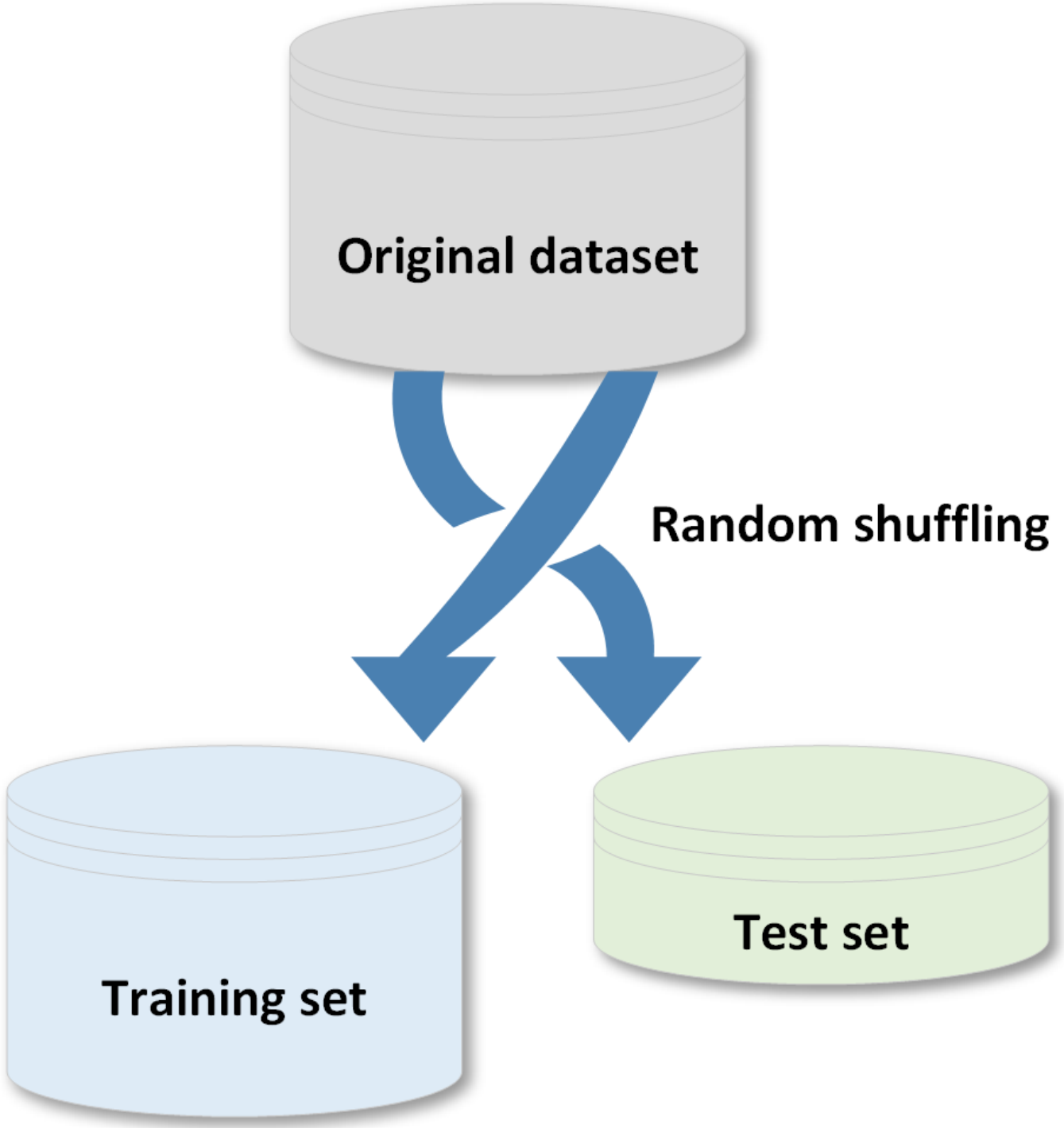
Creating training and test sets

When a dataset is large enough, it's a good practice to split it into training and test sets; the former to be used for training the model and the latter to test its performances. In the following figure, there's a schematic representation of this process:



There are two main rules in performing such an operation:

- Both datasets must reflect the original distribution
- The original dataset must be randomly shuffled before the split phase in order to avoid a correlation between consequent elements

With scikit-learn, this can be achieved using the `train_test_split()` function:

Copy

```
fromsklearn.model_selectionimporttrain_test_split

>>> X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=1000)
```

The parameter `test_size` (as well as `training_size`) allows specifying the percentage of elements to put into the test/training set. In this case, the ratio is 75 percent for training and 25 percent for the test phase. Another important parameter...

‹ Previous Section (/book/big_data_and_business_intelligence/9781785889622/3/ch03lvl1s

Next Section › (/book/big_data_and_business_intelligence/9781785889622/3/ch03lvl1sec2