

## ▼ LEAVE P OUT (LPO)

```
import numpy as np
from sklearn.model_selection import LeavePOut
```

```
X = np.ones(4)
lpo = LeavePOut(p=2)
for train, test in lpo.split(X):
    print("%s %s" % (train, test))
```

```
[2 3] [0 1]
[1 3] [0 2]
[1 2] [0 3]
[0 3] [1 2]
[0 2] [1 3]
[0 1] [2 3]
```

## ▼ Group K-Fold

```
from sklearn.model_selection import GroupKFold
```

```
X = [0.1, 0.2, 2.2, 2.4, 2.3, 4.55, 5.8, 8.8, 9, 10]
y = ["a", "b", "b", "b", "c", "c", "c", "d", "d", "d"]
groups = [1, 1, 1, 2, 2, 2, 3, 3, 3, 3]
```

```
gkf = GroupKFold(n_splits=3)
for train, test in gkf.split(X, y, groups=groups):
    print("%s %s" % (train, test))
```

```
[0 1 2 3 4 5] [6 7 8 9]
```

```
[0 1 2 6 7 8 9] [3 4 5]
[3 4 5 6 7 8 9] [0 1 2]
```

## ▼ Random permutations cross-validation a.k.a. Shuffle & Split

```
from sklearn.model_selection import ShuffleSplit
X = np.arange(10)
ss = ShuffleSplit(n_splits=5, test_size=0.25, random_state=
for train_index, test_index in ss.split(X):
    print("%s %s" % (train_index, test_index))
```

```
[9 1 6 7 3 0 5] [2 8 4]
[2 9 8 0 6 7 4] [3 5 1]
[4 5 1 0 6 9 7] [2 3 8]
[2 7 5 8 0 3 4] [6 1 9]
[4 1 0 6 8 9 3] [5 2 7]
```

## ▼ Leave One Group Out

```
from sklearn.model_selection import LeaveOneGroupOut

X = np.array([1, 5, 10, 50, 60, 70, 80])
y = [0, 1, 1, 2, 2, 2, 2]
groups = [1, 1, 2, 2, 3, 3, 3]
logo = LeaveOneGroupOut()
for train, test in logo.split(X, y, groups=groups):
    print("%s %s" % (train, test))
```

```
[2 3 4 5 6] [0 1]
```

```
↳ [0 1 4 5 6] [2 3]
    [0 1 2 3] [4 5 6]
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 1:32 AM

● ✕