hold-out-cv

In [3]:
```
import numpy as np
from sklearn.model_selection import train_test_split
import pandas as pd
```

In [4]:
```
x=[10,20,30,40,50,60,70,80,90,100]
```

In [5]:
```
x_train,x_test=train_test_split(x,test_size=0.3)
```

In [6]:
```
x_train
```

Out[6]:
```
[30, 80, 20, 10, 70, 100, 60]
```

In [7]:
```
x_test
```

Out[7]:
```
[90, 40, 50]
```

Leave-one-out-cv

In [10]:
```
from sklearn.model_selection import LeaveOneOut
```

In [11]:
```
l = LeaveOneOut()
```

In [13]:
```
for train,test in l.split(x):
    print(train,test)
```

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

Leave-P-Out-CV

In [16]:
```
from sklearn.model_selection import LeavePOut
x = np.array([[1,2],[3,4],[5,6],[7,8]])
y = np.array([1,2,3,4])
lpo = LeavePOut(2)
for train_index,test_index in lpo.split(x):
    print("TRAIN:",train_index,"TEST:",test_index)
    x_train,x_test = x[train_index],x[test_index]
    y_train,y_test=y[train_index],y[test_index]
```

```
TRAIN: [3] TEST: [0 1 2]
TRAIN: [2] TEST: [0 1 3]
TRAIN: [1] TEST: [0 2 3]
TRAIN: [0] TEST: [1 2 3]
```

K-fold cross validation

In [18]:
```
from sklearn.model_selection import KFold
x=["a","b","c","d","e","f"]
kf = KFold(n_splits=3,shuffle=False,random_state=None)
for train,test in kf.split(x):
    print("Train Data",train,"Test Data",test)
```

```
Train Data [2 3 4 5] Test Data [0 1]
Train Data [0 1 4 5] Test Data [2 3]
Train Data [0 1 2 3] Test Data [4 5]
```

Stratified K-Fold CV

In [19]:
```
from sklearn.model_selection import StratifiedKFold
x = np.array([[1,2],[3,4],[5,6],[7,8],[9,10],[11,12]])
y = np.array([0,0,1,0,1,1])

skf = StratifiedKFold(n_splits=3,random_state=None,shuffle=False)

for train_index,test_index in skf.split(x,y):
    print("Train:",train_index,"Test:",test_index)
    x_train,x_test = x[train_index],x[test_index]
    y_train,y_test=y[train_index],y[test_index]
```

```
Train: [1 3 4 5] Test: [0 2]
Train: [0 2 3 5] Test: [1 4]
Train: [0 1 2 4] Test: [3 5]
```

In [ ]:
```
```