Chapter 9: Demo Cross Validation

In [1]:

```
#Import scikit-learn dataset library
from sklearn import datasets

#Load dataset
iris = datasets.load_iris()
```

In [2]:

```
type(iris)
```

Out[2]:

```
sklearn.utils.Bunch
```

In [3]:

```
# print the label species(setosa, versicolor,virginica)
print(iris.target_names)

# print the names of the four features
print(iris.feature_names)
```

```
['setosa' 'versicolor' 'virginica']
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (c
m)']
```

In [4]:

```
# print the iris data (top 5 records)
print(iris.data[0:5])

# print the iris labels (0:setosa, 1:versicolor, 2:virginica)
print(iris.target[:5])
```

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
[0 0 0 0 0]
```

In [5]:

```
# Creating a DataFrame of given iris dataset.
import pandas as pd
data=pd.DataFrame({
    'sepal length':iris.data[:,0],
    'sepal width':iris.data[:,1],
    'petal length':iris.data[:,2],
    'petal width':iris.data[:,3],
    'species':iris.target
})
data.head()
```

Out[5]:

	sepal length	sepal width	petal length	petal width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [6]:

```
X=data[['petal length', 'petal width']]
y=data['species']
```

In [7]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

Cross Validation - Xác thực chéo

In [8]:

```
# 70%, 75%, 80% training and 30%, 25%, 25% test
test_size_lst = [0.3, 0.25, 0.2]
for i in test_size_lst:
    X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(X, y,
                                                        test_size=i)
    clf1=RandomForestClassifier(n_estimators=100)
    clf1.fit(X_train_1,y_train_1)

    score_train = clf1.score(X_train_1, y_train_1)
    score_test = clf1.score(X_test_1, y_test_1)

    print("With [", 1-i, ":", i, "], score train is ", round(score_train,2),
          ", score test is", round(score_test,2),
          "diff is", round(abs(score_train-score_test),2))
```

```
With [ 0.7 : 0.3 ], score train is  0.99 , score test is 0.98 diff is 0.01
With [ 0.75 : 0.25 ], score train is  1.0 , score test is 0.92 diff is 0.08
With [ 0.8 : 0.2 ], score train is  1.0 , score test is 0.93 diff is 0.07
```

In [9]:

```
# Compare: 70%-30%, 75%-25% and 80%-20%
# Choose the best one
# (Can run many times to make sure your choice)
```

k-folds

In [10]:
```
X.head()
```

Out[10]:

	petal length	petal width
0	1.4	0.2
1	1.4	0.2
2	1.3	0.2
3	1.5	0.2
4	1.4	0.2

In [11]:
```
y.values
```

Out[11]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [12]:
```
from sklearn import model_selection
from sklearn.model_selection import KFold
```

In [13]:
```
clf_k=RandomForestClassifier(n_estimators=100)
kfold = KFold(n_splits=10)
results = model_selection.cross_val_score(clf_k, X, y, cv=kfold)
print("Accuracy: %.2f%% (%.2f%%)" % (results.mean()*100.0,
                                     results.std()*100.0))
```

```
Accuracy: 93.33% (10.33%)
```

In [14]:
```
results
```

Out[14]:
```
array([1.        , 1.        , 1.        , 1.        , 0.93333333,
       0.86666667, 1.        , 0.86666667, 0.66666667, 1.        ])
```