In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import svm

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_sc
from sklearn.metrics import confusion_matrix
```

In [2]:
```
iris_data = datasets.load_iris()
```

In [3]:
```
iris = pd.DataFrame(iris_data.data, columns=iris_data.feature_names)
iris['target'] = iris_data.target
```

In [4]:
```
iris
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [5]:
```
#splitting data to training and testing dataset

# Input Data
x = iris_data.data

# Output Data
y = iris_data.target

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.2,
                                                random_state = 0)

print("xtrain shape : ", xtrain.shape)
print("xtest shape  : ", xtest.shape)
print("ytrain shape : ", ytrain.shape)
print("ytest shape  : ", ytest.shape)
```

```
xtrain shape :  (120, 4)
xtest shape  :  (30, 4)
ytrain shape :  (120,)
ytest shape  :  (30,)
```

In [8]:
```

clf =  svm.SVC()
clf.fit(xtrain, ytrain)

# predicting the test set results
y_pred = clf.predict(xtest)
```

In [9]:
```
print('Accuracy score: ', format(accuracy_score(ytest, y_pred)))
print('Precision score: ', format(precision_score(ytest, y_pred, average='micro'
print('Recall score: ', format(recall_score(ytest, y_pred, average='micro')))
print('F1 score: ', format(f1_score(ytest, y_pred, average='micro')))
print('\nConfusion Matrix :\n', confusion_matrix(ytest, y_pred))
```

```
Accuracy score:  1.0
Precision score:  1.0
Recall score:  1.0
F1 score:  1.0

Confusion Matrix :
 [[11  0  0]
 [ 0 13  0]
 [ 0  0  6]]
```