In [1]:
```
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.metrics import confusion_matrix, accuracy_score

from sklearn import svm
from sklearn.linear_model import LogisticRegression, SGDClassifier
```

In [2]:
```
data = load_iris()
x_data = data.data
y_data = data.target
x_train,x_test,y_train,y_test = train_test_split(x_data, y_data, test_size = 0.2(
```

SVC

In [3]:
```
clf =  svm.SVC()
clf.fit(x_train, y_train)

# predicting the test set results
y_pred = clf.predict(x_test)
print('\nConfusion Matrix :\n', confusion_matrix(y_test, y_pred))
print('Accuracy score: ', format(accuracy_score(y_test, y_pred)))
```

```
Confusion Matrix :
 [[ 8  0  0]
 [ 0 12  0]
 [ 0  1  9]]
Accuracy score:  0.9666666666666667
```

Logistic Regression

In [4]:
```
lr =  LogisticRegression()
lr.fit(x_train, y_train)

# predicting the test set results
y_pred = clf.predict(x_test)
print('\nConfusion Matrix :\n', confusion_matrix(y_test, y_pred))
print('Accuracy score: ', format(accuracy_score(y_test, y_pred)))
```

```
Confusion Matrix :
 [[ 8  0  0]
 [ 0 12  0]
 [ 0  1  9]]
Accuracy score:  0.9666666666666667
```

stochastic gradient descent

In [5]:
```
sgd = SGDClassifier(loss="hinge", penalty="l2", max_iter=1500)
sgd.fit(x_train, y_train)

# predicting the test set results
y_pred = sgd.predict(x_test)
print('\nConfusion Matrix :\n', confusion_matrix(y_test, y_pred))
print('Accuracy score: ', format(accuracy_score(y_test, y_pred)))
```

```
Confusion Matrix :
 [[ 8  0  0]
 [ 0 10  2]
 [ 0  0 10]]
Accuracy score:  0.9333333333333333
```