In [15]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
import numpy as np
```

In [16]:

```
irisData = load_iris()
```

In [17]:

```
#consist of four attributes ( attribute of specific type of iris plant)
print(irisData.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width
(cm)']
```

In [18]:

```
#task is to predict the classes which these plants belongs
print(irisData.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

In [19]:

```
# create feature and target arrays(split dataset into attributes and labels )
#x contain first four coloumns of dataset of all the attributes
 X= irisData.data
#y contains the labels
y= irisData.target
```

In [6]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=0)
```

In [30]:

```
#using k=2
knn=KNeighborsClassifier(n_neighbors=2)
knn.fit(X_train,y_train)
```

Out[30]:

```
KNeighborsClassifier(n_neighbors=2)
```

In [31]:

```
y_pred=knn.predict(X_test)
```

In [32]:

```
y_pred
```

Out[32]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0])
```

In [33]:

```
from sklearn.metrics import  classification_report ,confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[11  0  0]
 [ 0 13  0]
 [ 0  1  5]]
[[11  0  0]
 [ 0 13  0]
 [ 0  1  5]]
```

In [34]:

```
# Evaluating performance metrics

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Accuracy
print("\nAccuracy: ", accuracy_score(y_test, y_pred))

# Precision
print("\nPrecision: ", precision_score(y_test, y_pred, average='weighted', zero_division=1))

# Recall
print("\nRecall: ", recall_score(y_test,y_pred, average='weighted'))

# F-1 Score
print("\nF1 score: ", f1_score(y_test, y_pred, average='weighted'))
```

```
Accuracy:  0.9666666666666667

Precision:  0.9690476190476189

Recall:  0.9666666666666667

F1 score:  0.9657687991021324
```

In [ ]:

```

```