In [26]:

```
import numpy as np
import pandas as pd
```

In [27]:

```
df=pd.read_csv("/home/aparna/Downloads/Iris.csv")
```

In [28]:

```
df=df.drop(['Id'],axis=1)
```

In [29]:

```
print(df)
```

```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm         Species
0              5.1           3.5            1.4           0.2     Iris-setosa
1              4.9           3.0            1.4           0.2     Iris-setosa
2              4.7           3.2            1.3           0.2     Iris-setosa
3              4.6           3.1            1.5           0.2     Iris-setosa
4              5.0           3.6            1.4           0.2     Iris-setosa
..             ...           ...            ...           ...             ...
145            6.7           3.0            5.2           2.3  Iris-virginica
146            6.3           2.5            5.0           1.9  Iris-virginica
147            6.5           3.0            5.2           2.0  Iris-virginica
148            6.2           3.4            5.4           2.3  Iris-virginica
149            5.9           3.0            5.1           1.8  Iris-virginica

[150 rows x 5 columns]
```

In [30]:

```
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
```

In [31]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

In [32]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

In [33]:

```
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train,y_train)
```

Out[33]:

```
▼ KNeighborsClassifier
KNeighborsClassifier()
```

In [34]:

```
y_pred=classifier.predict(x_test)
print(classifier.predict([[5.1,3.8,1.6,0.4]]))
```

```
['Iris-virginica']
```

In [37]:

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
result=confusion_matrix(y_test,y_pred)
print(result)
result1=accuracy_score(y_test,y_pred)
print(result1)
```

```
[[13  0  0]
 [ 0 15  1]
 [ 0  0 16]]
0.9777777777777777
```

In [ ]: