In [1]:

```python
import sklearn

import pandas as pd

from sklearn.datasets import load_iris

iris=load_iris()

iris.keys()

df=pd.DataFrame(iris['data'])

print(df)

print(iris['target_names'])

iris['feature_names']
```

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

[150 rows x 4 columns]
['setosa' 'versicolor' 'virginica']
```

Out[1]:

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

In [2]:

```python
X=df

y=iris['target']
```

In [3]:

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

In [4]:

```python
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=3)
```

In [5]:

```python
knn.fit(X_train,y_train)
```

Out[5]:

```
KNeighborsClassifier(n_neighbors=3)
```

In [6]:

```python
import numpy as np

x_new=np.array([[5,2.9,1,0.2]])
```

In [7]:

```python
prediction=knn.predict(x_new)

iris['target_names'][prediction]
```

Out[7]:

```
array(['setosa'], dtype='<U10')
```

In [8]:

```python
from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

y_pred=knn.predict(X_test)

cm=confusion_matrix(y_test,y_pred)

print(cm)

print(" correct predicition",accuracy_score(y_test,y_pred))

print(" worng predicition",(1-accuracy_score(y_test,y_pred)))
```

```
[[19  0  0]
 [ 0 15  0]
 [ 0  1 15]]
 correct predicition 0.98
 worng predicition 0.020000000000000018
```

In [ ]: