

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

```
In [4]: df = pd.read_csv('iris_csv.csv')
df
```

```
Out[4]:
```

	sepalength	sepalwidth	petallength	petalwidth	class
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 × 5 columns

```
In [5]: df.isnull().sum()
```

```
Out[5]: sepalength    0
sepalwidth    0
petallength    0
petalwidth    0
class         0
dtype: int64
```

```
In [6]: df.shape
```

```
Out[6]: (150, 5)
```

```
In [7]: df.describe()
```

```
Out[7]:
```

	sepalength	sepalwidth	petallength	petalwidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
In [13]: x = df.iloc[:, :-1]  
y = df.iloc[:, -1:]
```

```
In [18]: from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_stat
```

```
In [19]: x_train.shape
```

```
Out[19]: (120, 4)
```

```
In [20]: x_test.shape
```

```
Out[20]: (30, 4)
```

```
In [21]: from sklearn.preprocessing import StandardScaler
```

```
In [24]: sc = StandardScaler()  
x_train = sc.fit_transform(x_train)  
x_test = sc.fit_transform(x_test)
```

```
In [26]: from sklearn.neighbors import KNeighborsClassifier  
classifier = KNeighborsClassifier()
```

```
In [27]: classifier.fit(x_train,y_train)
```

```
C:\Users\prati\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:215: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
```

```
    return self._fit(X, y)
```

```
Out[27]: KNeighborsClassifier()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**

**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [28]: y_pred = classifier.predict(x_test)
```

```
In [31]: from sklearn.metrics import accuracy_score, confusion_matrix
accuracy_score(y_test,y_pred)
```

```
Out[31]: 0.8666666666666667
```

```
In [32]: confusion_matrix(y_test,y_pred)
```

```
Out[32]: array([[11,  0,  0],
                [ 0,  9,  4],
                [ 0,  0,  6]], dtype=int64)
```

```
In [ ]:
```