In [2]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_csv("shirts.csv")
df.head()
```

Out[3]:

	Height	Weight	Size
0	158	58	M
1	158	59	M
2	158	63	M
3	160	59	M
4	160	60	M

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Height    0
Weight    0
Size      0
dtype: int64
```

In [5]:
```
df.dtypes
```

Out[5]:
```
Height     int64
Weight     int64
Size      object
dtype: object
```

In [6]:
```
x = df[["Height","Weight"]]
x.head()
```

Out[6]:

	Height	Weight
0	158	58
1	158	59
2	158	63
3	160	59
4	160	60

In [7]:
```
# target
y = df["Size"]
y.head()
```

Out[7]:
```
0    M
1    M
2    M
3    M
4    M
Name: Size, dtype: object
```

In [9]:
```
y1 = pd.get_dummies(df["Size"])
y1
```

Out[9]:

	L	M
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	1	0
8	1	0
9	1	0
10	1	0
11	1	0
12	1	0
13	1	0
14	1	0
15	1	0
16	1	0
17	1	0

In [8]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [12]:
```
knn = KNeighborsClassifier()
```

In [13]:
```
knn.fit(x,y1["L"])
```

Out[13]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=5, p=2,
           weights='uniform')
```

In [14]:
```
# test the model
y_pred = knn.predict(x)
y_pred
```

Out[14]:
```
array([0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=uint8)
```

In [7]:
```
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [16]:
```
accuracy_score(y1["L"],y_pred)
```

Out[16]:
```
0.8333333333333334
```

In [17]:
```
confusion_matrix(y1["L"],y_pred)
```

Out[17]:
```
array([[ 5,  2],
       [ 1, 10]], dtype=int64)
```

In [18]:
```
df.shape
```

Out[18]:
```
(18, 3)
```

In [19]:
```
knn.predict([[158,58]])
```

Out[19]:
```
array([0], dtype=uint8)
```

In [20]:
```
#accuracy = (TP+TN)/(TP+TN+FP+FN)
(5+10)/(5+10+2+1)
```

Out[20]:
```
0.8333333333333334
```

In [ ]:
```

```

In [14]:
```
from sklearn.datasets import load_iris
```

In [15]:
```
iris_data = load_iris()
iris_data
```

Out[15]:
```
{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
        [5.1, 3.5, 1.4, 0.3],
        [5.7, 3.8, 1.7, 0.3],
```

In [16]:
```
x = iris_data["data"]
x
```

Out[16]:
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
```

In [17]:
```
y = iris_data["target"]
y
```

Out[17]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [18]:
```
import pandas as pd
y1 = pd.get_dummies(iris_data.target)
y1
```

Out[18]:

	0	1	2
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	1	0	0
11	1	0	0
12	1	0	0
13	1	0	0
14	1	0	0
15	1	0	0
16	1	0	0
17	1	0	0
18	1	0	0
19	1	0	0
20	1	0	0
21	1	0	0
22	1	0	0
23	1	0	0
24	1	0	0
25	1	0	0
26	1	0	0
27	1	0	0
28	1	0	0
29	1	0	0
...	...	...	...
120	0	0	1
121	0	0	1

	0	1	2
122	0	0	1
123	0	0	1
124	0	0	1
125	0	0	1
126	0	0	1
127	0	0	1
128	0	0	1
129	0	0	1
130	0	0	1
131	0	0	1
132	0	0	1
133	0	0	1
134	0	0	1
135	0	0	1
136	0	0	1
137	0	0	1
138	0	0	1
139	0	0	1
140	0	0	1
141	0	0	1
142	0	0	1
143	0	0	1
144	0	0	1
145	0	0	1
146	0	0	1
147	0	0	1
148	0	0	1
149	0	0	1

150 rows × 3 columns

In [25]:
```
kn1 = KNeighborsClassifier(n_neighbors=2)
```

In [26]:
```
kn1.fit(x,y1)
```

Out[26]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=2, p=2,
                     weights='uniform')
```

In [27]:
```
y_pred1 = kn1.predict(x)
y_pred1
```

Out[27]:
```
array([[1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
       [1, 0, 0],
```

In [28]:
```
accuracy_score(y1,y_pred1)
```

Out[28]:
```
0.96
```

In [29]:
```
kn1.predict([[5.1, 3.5, 1.4, 0.2]])
```

Out[29]:
```
array([[1, 0, 0]], dtype=uint8)
```

In [24]:
```
#neghbiors =2
```

In [ ]: