In [1]:

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

In [2]:

```
df=pd.read_csv(r"https://github.com/YBI-Foundation/Dataset/raw/main/Fruits.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Fruit Category	Fruit Name	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
0	1	Apple	192	8.4	7.3	0.55
1	1	Apple	180	8.0	6.8	0.59
2	1	Apple	176	7.4	7.2	0.60
3	1	Apple	178	7.1	7.8	0.92
4	1	Apple	172	7.4	7.0	0.89

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59 entries, 0 to 58
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Fruit Category      59 non-null     int64  
 1   Fruit Name          59 non-null     object 
 2   Fruit Weight        59 non-null     int64  
 3   Fruit Width         59 non-null     float64
 4   Fruit Length        59 non-null     float64
 5   Fruit Colour Score  59 non-null     float64
dtypes: float64(3), int64(2), object(1)
memory usage: 2.9+ KB
```

In [5]:

```
df.describe()
```

Out[5]:

	Fruit Category	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
count	59.000000	59.000000	59.000000	59.000000	59.000000
mean	1.949153	141.796610	7.105085	7.693220	0.762881
std	0.775125	67.335951	0.816938	1.361017	0.076857
min	1.000000	58.000000	5.800000	4.000000	0.550000
25%	1.000000	82.000000	6.600000	7.200000	0.720000
50%	2.000000	154.000000	7.200000	7.600000	0.750000
75%	3.000000	167.000000	7.500000	8.200000	0.810000
max	3.000000	362.000000	9.600000	10.500000	0.930000

In [6]:

```
df.shape
```

Out[6]:

```
(59, 6)
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Fruit Category', 'Fruit Name', 'Fruit Weight', 'Fruit Width',
       'Fruit Length', 'Fruit Colour Score'],
      dtype='object')
```

In [8]:

```
df['Fruit Category'].value_counts()
```

Out[8]:

```
2    24
1    19
3    16
Name: Fruit Category, dtype: int64
```

In [9]:

```
df['Fruit Category'].mean()
```

Out[9]:

```
1.9491525423728813
```

In [10]:

```
y=df['Fruit Category']
```

In [11]:

```
y.shape
```

Out[11]:

```
(59,)
```

In [12]:

```
y
```

Out[12]:

```
0     1
1     1
2     1
3     1
4     1
5     1
6     1
7     1
8     1
9     1
10    1
11    1
12    1
13    1
14    1
15    1
16    1
17    1
18    1
19    2
20    2
21    2
22    2
23    2
24    2
25    2
26    2
27    2
28    2
29    2
30    2
31    2
32    2
33    2
34    2
35    2
36    2
37    2
38    2
39    2
40    2
41    2
42    2
43    3
44    3
45    3
46    3
47    3
48    3
49    3
50    3
51    3
52    3
53    3
```

```
54    3
55    3
56    3
57    3
58    3
Name: Fruit Category, dtype: int64
```

In [13]:

```
x=df[['Fruit Weight','Fruit Width','Fruit Length','Fruit Colour Score']]
```

In [14]:

```
x=df.drop(['Fruit Category','Fruit Name'],axis=1)
```

In [15]:

```
x.shape
```

Out[15]:

```
(59, 4)
```

In [16]:

```
x
```

Out[16]:

	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
0	192	8.4	7.3	0.55
1	180	8.0	6.8	0.59
2	176	7.4	7.2	0.60
3	178	7.1	7.8	0.92
4	172	7.4	7.0	0.89
5	166	6.9	7.3	0.93
6	172	7.1	7.6	0.92
7	154	7.0	7.1	0.88
8	164	7.3	7.7	0.70
9	152	7.6	7.3	0.69
10	156	7.7	7.1	0.69
11	156	7.6	7.5	0.67
12	168	7.5	7.6	0.73
13	162	7.5	7.1	0.83
14	162	7.4	7.2	0.85
15	160	7.5	7.5	0.86
16	156	7.4	7.4	0.84
17	140	7.3	7.1	0.87
18	170	7.6	7.9	0.88
19	86	6.2	4.7	0.80
20	84	6.0	4.6	0.79
21	80	5.8	4.3	0.77
22	80	5.9	4.3	0.81
23	76	5.8	4.0	0.81
24	342	9.0	9.4	0.75
25	356	9.2	9.2	0.75
26	362	9.6	9.2	0.74
27	204	7.5	9.2	0.77
28	140	6.7	7.1	0.72
29	160	7.0	7.4	0.81
30	158	7.1	7.5	0.79
31	210	7.8	8.0	0.82
32	164	7.2	7.0	0.80
33	190	7.5	8.1	0.74

	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
34	142	7.6	7.8	0.75
35	150	7.1	7.9	0.75
36	160	7.1	7.6	0.76
37	154	7.3	7.3	0.79
38	158	7.2	7.8	0.77
39	144	6.8	7.4	0.75
40	154	7.1	7.5	0.78
41	180	7.6	8.2	0.79
42	154	7.2	7.2	0.82
43	97	7.2	10.3	0.70
44	70	7.3	10.5	0.72
45	93	7.2	9.2	0.72
46	80	7.3	10.2	0.71
47	98	7.3	9.7	0.72
48	87	7.3	10.1	0.72
49	66	5.8	8.7	0.73
50	65	6.0	8.2	0.71
51	58	6.0	7.5	0.72
52	59	5.9	8.0	0.72
53	60	6.0	8.4	0.74
54	58	6.1	8.5	0.71
55	58	6.3	7.7	0.72
56	58	5.9	8.1	0.73
57	76	6.5	8.5	0.72
58	59	6.1	8.1	0.70

In [17]:

```
from sklearn.model_selection import train_test_split
```

In [18]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=2529)
```

In [19]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[19]:

```
((41, 4), (18, 4), (41,), (18,))
```

In [20]:

```
from sklearn.linear_model import LogisticRegression
```

In [21]:

```
model=LogisticRegression(max_iter=500)
```

In [22]:

```
model.fit(x_train,y_train)
```

Out[22]:

```
LogisticRegression(max_iter=500)
```

In [23]:

```
y_pred=model.predict(x_test)
```

In [24]:

```
y_pred.shape
```

Out[24]:

```
(18,)
```

In [25]:

```
y_pred
```

Out[25]:

```
array([3, 3, 2, 2, 2, 2, 2, 3, 2, 1, 3, 1, 2, 3, 1, 2, 1, 3], dtype=int64)
```

In [26]:

```
model.predict_proba(x_test)
```

Out[26]:

```
array([[2.41440823e-03, 3.19602832e-03, 9.94389563e-01],
       [2.13636686e-03, 2.41088316e-03, 9.95452750e-01],
       [4.00713921e-01, 5.98801995e-01, 4.84084907e-04],
       [4.38508421e-01, 5.61489561e-01, 2.01838911e-06],
       [4.08812155e-01, 5.91163054e-01, 2.47913524e-05],
       [4.53047378e-01, 5.46412072e-01, 5.40550585e-04],
       [2.52140320e-01, 7.01792506e-01, 4.60671740e-02],
       [8.65690885e-03, 1.85201915e-03, 9.89491072e-01],
       [4.95658388e-01, 5.04155906e-01, 1.85706289e-04],
       [6.04386045e-01, 3.95200213e-01, 4.13741999e-04],
       [1.42148981e-02, 1.02038825e-02, 9.75581219e-01],
       [5.75888601e-01, 4.23396386e-01, 7.15013131e-04],
       [4.17772607e-01, 5.82227393e-01, 1.38895267e-13],
       [7.45978364e-04, 1.09458392e-04, 9.99144563e-01],
       [6.72406526e-01, 3.27585801e-01, 7.67370983e-06],
       [3.63778424e-01, 6.36123746e-01, 9.78292814e-05],
       [5.76163921e-01, 4.23820424e-01, 1.56544095e-05],
       [5.00270216e-03, 3.81395515e-03, 9.91183343e-01]])
```

In [27]:

```
from sklearn.metrics import confusion_matrix,classification_report
```

In [28]:

```
print(confusion_matrix(y_test,y_pred))
```

```
[[4 2 0]
 [0 6 0]
 [0 0 6]]
```

In [29]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         6
           2       0.75      1.00      0.86         6
           3       1.00      1.00      1.00         6

    accuracy                           0.89        18
   macro avg       0.92      0.89      0.89        18
weighted avg       0.92      0.89      0.89        18
```

In [30]:

```
df_new=df.sample(1)
```

In [31]:

```
df_new
```

Out[31]:

	Fruit Category	Fruit Name	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
21	2	Orange	80	5.8	4.3	0.77

In [32]:

```
x_new=df_new[['Fruit Weight','Fruit Width','Fruit Length','Fruit Colour Score']]
```

In [33]:

```
x_new.shape
```

Out[33]:

```
(1, 4)
```

In [34]:

```
y_pred_new=model.predict(x_new)
```

In [35]:

```
y_pred_new
```

Out[35]:

```
array([2], dtype=int64)
```

In [36]:

```
model.predict_proba(x_new)
```

Out[36]:

```
array([[0.2040408 , 0.75236662, 0.04359259]])
```