In [1]:
```python
import pandas as pd
import numpy as np
```

In [3]:
```python
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Fruits.csv')
```

In [4]:
```python
df.head()
```

Out[4]:

	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 [5]:
```python
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 [6]:
```python
df.describe()
```

Out[6]:

	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 [7]:
```python
df.columns
```

Out[7]:
```
Index(['Fruit Category', 'Fruit Name', 'Fruit Weight', 'Fruit Width',
       'Fruit Length', 'Fruit Colour Score'],
      dtype='object')
```

In [8]:
```python
df['Fruit Category'].value_counts()
```

Out[8]:
```
2    24
1    19
3    16
Name: Fruit Category, dtype: int64
```

In [11]:
```python
df.groupby('Fruit Category').mean()
```

Out[11]:

	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
Fruit Category				
1	165.052632	7.457895	7.342105	0.783684
2	170.333333	7.220833	7.195833	0.776250
3	71.375000	6.512500	8.856250	0.718125

In [12]:
```python
y= df['Fruit Category']
```

In [13]:
```python
y.shape
```

Out[13]:
```
(59,)
```

In [14]:
```python
y
```

Out[14]:
```
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 [16]:
```python
X = df[['Fruit Weight','Fruit Width','Fruit Length','Fruit Colour Score']]
```

In [17]:
```python
X =df.drop(['Fruit Category','Fruit Name'],axis =1)
```

In [18]:
```python
X.shape
```

Out[18]:
```
(59, 4)
```

In [19]:
```python
X
```

Out[19]:

	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
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 [20]:
```python
from sklearn.model_selection import train_test_split
```

In [21]:
```python
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size= 0.3,stratify =y,random_state =2529)
```

In [22]:
```python
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

Out[22]:
```
((41, 4), (18, 4), (41,), (18,))
```

In [23]:
```python
from sklearn.linear_model import LogisticRegression
```

In [24]:
```python
model = LogisticRegression(max_iter= 500)
```

In [25]:
```python
model.fit(X_train,y_train)
```

Out[25]:
```
LogisticRegression(max_iter=500)
```

In [26]:
```python
y_pred = model.predict(X_test)
```

In [27]:
```python
y_pred.shape
```

Out[27]:
```
(18,)
```

In [28]:
```python
y_pred
```

Out[28]:
```
array([2, 3, 2, 2, 1, 3, 3, 3, 2, 2, 1, 1, 3, 1, 2, 2, 2, 2])
```

In [30]:
```python
model.predict_proba(X_test)
```

Out[30]:
```
array([[3.84869552e-01, 6.15130448e-01, 1.08223557e-13],
       [4.56010235e-03, 3.49593156e-03, 9.91943966e-01],
       [4.90244803e-01, 5.09657249e-01, 9.79484672e-05],
       [3.53486534e-01, 6.46199953e-01, 3.13512815e-04],
       [5.00285500e-01, 4.99528878e-01, 1.85622208e-04],
       [2.78142750e-03, 4.11007441e-03, 9.93108498e-01],
       [6.66174744e-04, 1.25563556e-04, 9.99208262e-01],
       [2.27546095e-03, 4.46885442e-03, 9.93255685e-01],
       [4.16180161e-01, 5.83016945e-01, 8.02893642e-04],
       [4.37039107e-01, 5.62634133e-01, 3.26759453e-04],
       [5.55463131e-01, 4.44104226e-01, 4.32642560e-04],
       [7.05089995e-01, 2.94902488e-01, 7.51619085e-06],
       [1.75178402e-02, 6.71863335e-03, 9.75763526e-01],
       [5.73018644e-01, 4.24186637e-01, 2.79471856e-03],
       [2.87935997e-01, 6.51887400e-01, 6.01766029e-02],
       [2.54284531e-01, 6.89224785e-01, 5.64906836e-02],
       [4.58172610e-01, 5.41779302e-01, 4.80886210e-05],
       [4.59473822e-01, 5.39971389e-01, 5.54788956e-04]])
```

In [31]:
```python
from sklearn.metrics import confusion_matrix,classification_report
```

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

```
[[4 2 0]
 [0 7 0]
 [0 0 5]]
```

In [33]:
```python
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         6
           2       0.78      1.00      0.88         7
           3       1.00      1.00      1.00         5

    accuracy                           0.89        18
   macro avg       0.93      0.89      0.89        18
weighted avg       0.91      0.89      0.88        18
```

In [34]:
```python
df_new =df.sample(1)
```

In [35]:
```python
df_new
```

Out[35]:

	Fruit Category	Fruit Name	Fruit Weight	Fruit Width	Fruit Length	Fruit Colour Score
5	1	Apple	166	6.9	7.3	0.93

In [36]:
```python
X_new = df_new[['Fruit Weight','Fruit Width','Fruit Length','Fruit Colour Score']]
```

In [37]:
```python
X_new.shape
```

Out[37]:
```
(1, 4)
```

In [38]:
```python
y_pred_new =model.predict(X_new)
```

In [39]:
```python
y_pred_new
```

Out[39]:
```
array([2])
```

In [41]:
```python
model.predict_proba(X_new)
```

Out[41]:
```
array([[3.04702405e-01, 6.95176276e-01, 1.21319460e-04]])
```

In [ ]: