import pandas as pd  
import numpy as np  
df=pd.read\_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Fruits.csv')  
df.head()

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

df.shape

(59, 6)

df.describe()

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

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

df.columns

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

df['Fruit Category'].value\_counts()

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

df.groupby('Fruit Category').mean()

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

y=df['Fruit Category']  
y.shape

(59,)

y

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

X=df[['Fruit Weight', 'Fruit Width','Fruit Length', 'Fruit Colour Score']]  
X.shape

(59, 4)

X

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

from sklearn.model\_selection import train\_test\_split  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size = 0.3, random\_state=2529)  
X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape

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

from sklearn.linear\_model import LogisticRegression   
model = LogisticRegression()  
model.fit(X\_train, y\_train)  
y\_pred= model.predict(X\_test)  
y\_pred.shape

D:\anaconda\lib\site-packages\sklearn\linear\_model\\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.  
  
Increase the number of iterations (max\_iter) or scale the data as shown in:  
 https://scikit-learn.org/stable/modules/preprocessing.html  
Please also refer to the documentation for alternative solver options:  
 https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression  
 n\_iter\_i = \_check\_optimize\_result(

(18,)

y\_pred

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

model.predict\_proba(X\_test)

array([[2.22618001e-03, 2.93433587e-03, 9.94839484e-01],  
 [1.99112268e-03, 2.24226575e-03, 9.95766612e-01],  
 [4.01467880e-01, 5.98240061e-01, 2.92059248e-04],  
 [4.39090169e-01, 5.60908950e-01, 8.80908455e-07],  
 [4.04669014e-01, 5.95318944e-01, 1.20418469e-05],  
 [4.51367001e-01, 5.48288889e-01, 3.44109721e-04],  
 [2.53002060e-01, 7.00235519e-01, 4.67624206e-02],  
 [8.38088868e-03, 2.00741549e-03, 9.89611696e-01],  
 [4.81383015e-01, 5.18508389e-01, 1.08595932e-04],  
 [5.75129339e-01, 4.24591113e-01, 2.79547964e-04],  
 [1.37567431e-02, 1.03464012e-02, 9.75896856e-01],  
 [5.48613531e-01, 4.50917863e-01, 4.68606490e-04],  
 [4.08523957e-01, 5.91476043e-01, 2.28128071e-14],  
 [6.36743052e-04, 1.06482117e-04, 9.99256775e-01],  
 [6.23356151e-01, 3.76639499e-01, 4.35010941e-06],  
 [3.81326896e-01, 6.18619555e-01, 5.35489925e-05],  
 [5.33959757e-01, 4.66031048e-01, 9.19544317e-06],  
 [4.20063281e-03, 3.38610595e-03, 9.92413261e-01]])

from sklearn.metrics import confusion\_matrix, classification\_report  
print(confusion\_matrix(y\_test,y\_pred))

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

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

df\_new=df.sample(1)  
df\_new

Fruit Category Fruit Name Fruit Weight Fruit Width Fruit Length \  
36 2 Orange 160 7.1 7.6   
  
 Fruit Colour Score   
36 0.76

X\_new=df\_new[['Fruit Weight', 'Fruit Width','Fruit Length', 'Fruit Colour Score']]  
X\_new.shape

(1, 4)

y\_pred\_new = model.predict(X\_new)  
y\_pred\_new

array([2], dtype=int64)

model.predict\_proba(X\_new)

array([[3.87941006e-01, 6.11802401e-01, 2.56593120e-04]])