```python
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
        from sklearn.model_selection import train_test_split
        from sklearn.naive_bayes import GaussianNB
        from sklearn.metrics import accuracy_score, classification_report
```

```python
In [2]: df=pd.read_csv('fruits.csv')
        df.head()
```

Out[2]:

	weight	size	color_score	Fruit_label
0	150	7.5	0.50	lemon
1	170	7.5	0.50	lemon
2	190	8.0	0.74	orange
3	210	8.5	0.75	orange
4	230	8.5	0.75	orange

```python
In [3]: x=df[['weight','size','color_score']]
        y=df['Fruit_label']
```

```python
In [4]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_sta
        clf=GaussianNB()
        clf.fit(X_train, y_train)
```

Out[4]: ▾ GaussianNB ⓘ ❓

```
GaussianNB()
```

```python
In [5]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_sta
        clf=GaussianNB()
        clf.fit(X_train, y_train)
```

Out[5]: ▾ GaussianNB ⓘ ❓

```
GaussianNB()
```

```python
In [6]: y_pred = clf.predict(X_test)
        print(classification_report(y_test, y_pred))
```
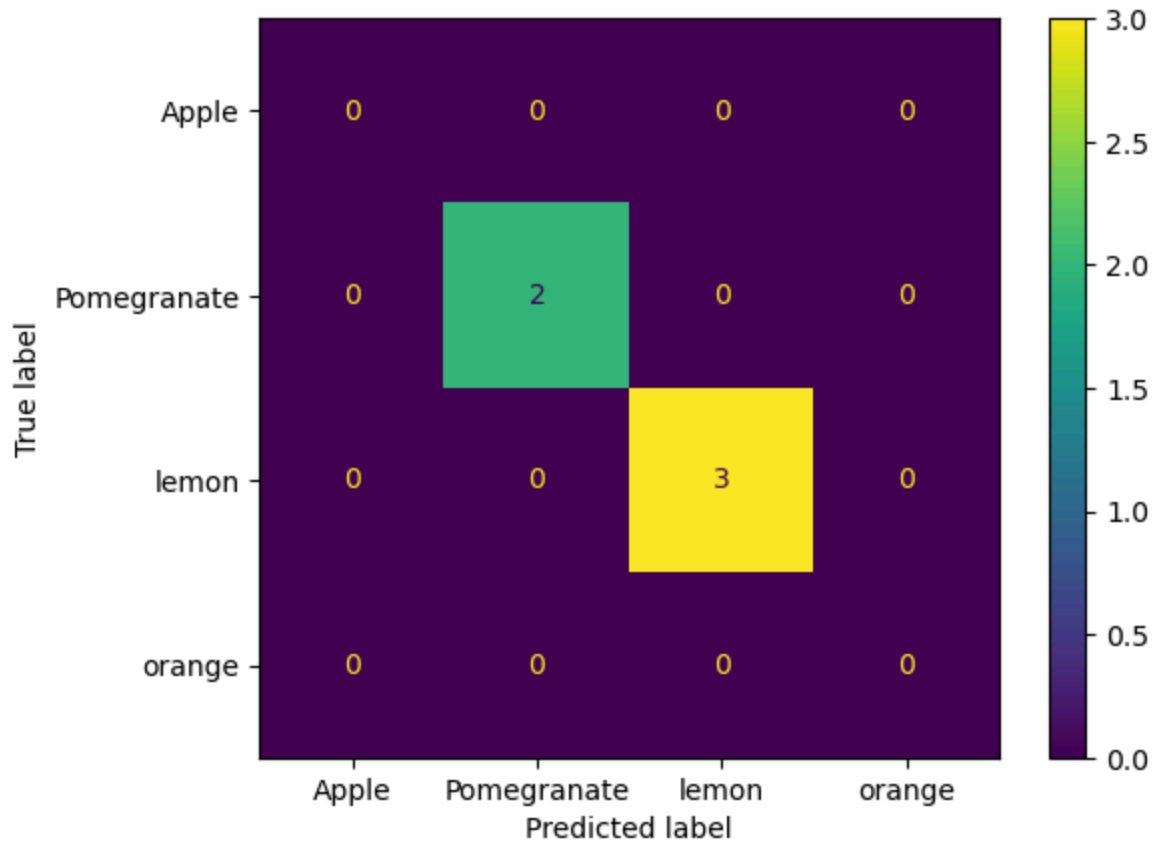
	precision	recall	f1-score	support
Pomegranate	1.00	1.00	1.00	2
lemon	1.00	1.00	1.00	3
accuracy			1.00	5
macro avg	1.00	1.00	1.00	5
weighted avg	1.00	1.00	1.00	5

In [7]:
```python
print(clf.predict([[300,9,0.9]]))
```

['Pomegranate']

In [9]:
```python
cm=confusion_matrix(y_test,y_pred,labels=clf.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=clf.classes_)
disp.plot();
```


In [20]:
```python
df=pd.read_csv('NBUser_data.csv')
df.head()
```

Out[20]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [22]:
```python
x=df[['Age','EstimatedSalary']]
y=df['Purchased']
```

In [24]:
```python
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_sta
clf=GaussianNB()
clf.fit(X_train, y_train)
```

Out[24]:
```
▾  GaussianNB  ⓘ ⓘ

GaussianNB()
```

In [26]:
```python
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94        73
           1       0.97      0.83      0.90        47

    accuracy                           0.93       120
   macro avg       0.94      0.91      0.92       120
weighted avg       0.93      0.93      0.92       120
```
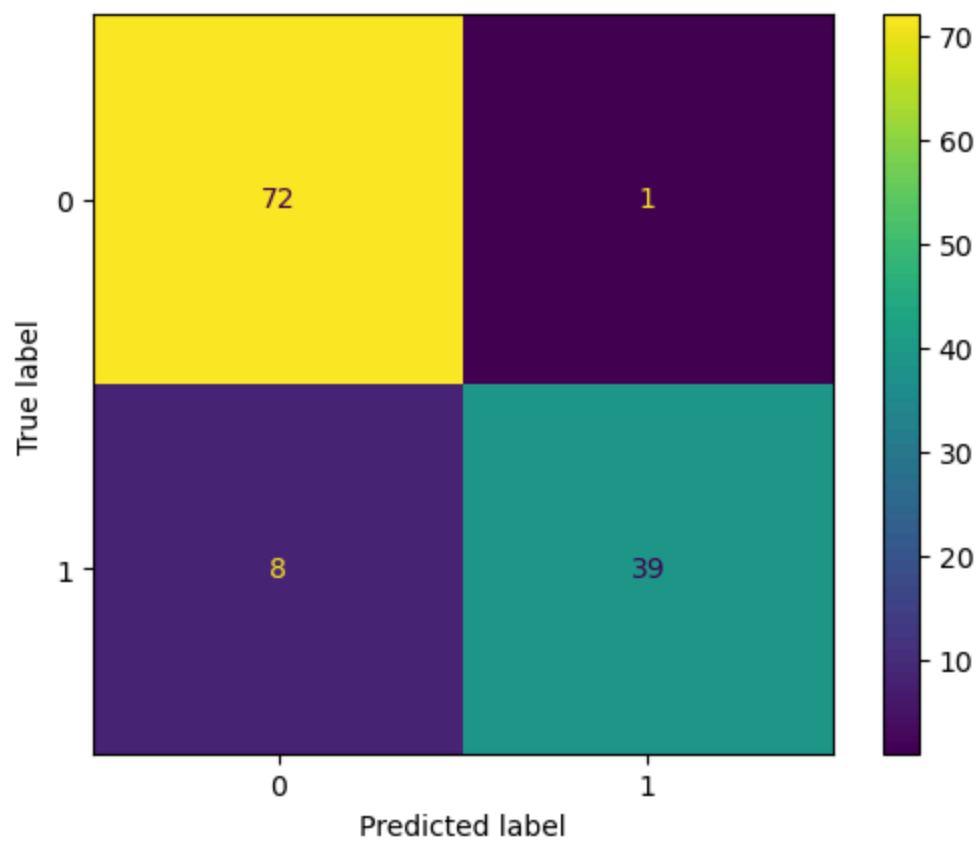
In [30]:
```python
print(clf.predict([[26,43000]]))
```

```
[0]
```
```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does
not have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```

In [32]:
```python
cm=confusion_matrix(y_test,y_pred,labels=clf.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=clf.classes_)
disp.plot();
```