In [22]:

```
'''
Patrick Ballou
ID: 801130521
ECGR 4105
Homework 3
Problem 1
'''
```

Out[22]:

```
'\nPatrick Ballou\nID: 801130521\nECGR 4105\nHomework 3\nProblem 1\n'
```

In [23]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
from sklearn.datasets import load_breast_cancer
from sklearn import datasets
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import PrecisionRecallDisplay
```

In [24]:

```
breast = load_breast_cancer()
breast_data = breast.data
breast_data.shape
breast_input = pd.DataFrame(breast_data)
```

In [25]:

```
breast_labels = breast.target
breast_labels.shape
labels = np.reshape(breast_labels,(569,1))
final_breast_data = np.concatenate([breast_data, labels],axis=1)
final_breast_data.shape
```

Out[25]:

```
(569, 31)
```

In [26]:

```
breast_dataset = pd.DataFrame(final_breast_data)
features = breast.feature_names
features
```

Out[26]:

```
array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

In [27]:

```
features_labels = np.append(features, 'label')
breast_dataset.columns = features_labels
breast_dataset.head()
```

Out[27]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	dim
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	(
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	(
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	(
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	(
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	(

5 rows × 31 columns

In [28]:

```
x_unstandardized = breast_dataset[features]
Y = breast_dataset['label']
```

In [30]:

```
#standard scaler is best here
scaler = StandardScaler()
#scaler = MinMaxScaler()
X = scaler.fit_transform(x_unstandardized)
model = GaussianNB()
X #final input matrix
```

Out[30]:

```
array([[ 1.09706398, -2.07333501,  1.26993369, ...,  2.29607613,
         2.75062224,  1.93701461],
       [ 1.82982061, -0.35363241,  1.68595471, ...,  1.0870843 ,
        -0.24388967,  0.28118999],
       [ 1.57988811,  0.45618695,  1.56650313, ...,  1.95500035,
         1.152255  ,  0.20139121],
       ...,
       [ 0.70228425,  2.0455738 ,  0.67267578, ...,  0.41406869,
        -1.10454895, -0.31840916],
       [ 1.83834103,  2.33645719,  1.98252415, ...,  2.28998549,
         1.91908301,  2.21963528],
       [-1.80840125,  1.22179204, -1.81438851, ..., -1.74506282,
        -0.04813821, -0.75120669]])
```

In [31]:

```
model.fit(X, Y)
```

Out[31]:

```
▾ GaussianNB
GaussianNB()
```

In [32]:

```
expected = Y
predicted = model.predict(X)
```

In [33]:

```
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
plt.rcParams["figure.figsize"] = (12,8)
```

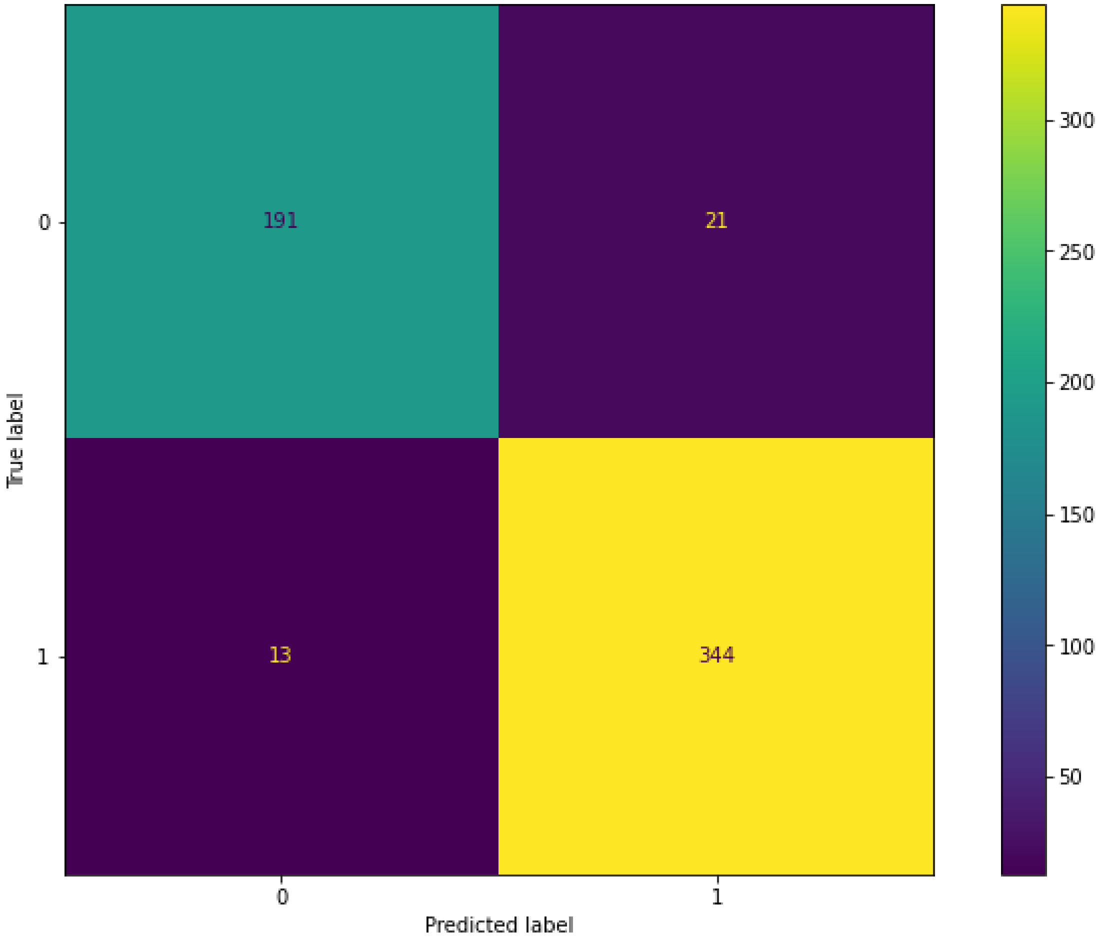
```
              precision    recall  f1-score   support

         0.0       0.94      0.90      0.92       212
         1.0       0.94      0.96      0.95       357

    accuracy                           0.94       569
   macro avg       0.94      0.93      0.94       569
weighted avg       0.94      0.94      0.94       569

[[191  21]
 [ 13 344]]
```

In [34]:

```
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix=metrics.confusion_matrix(
cm_display.plot()
```

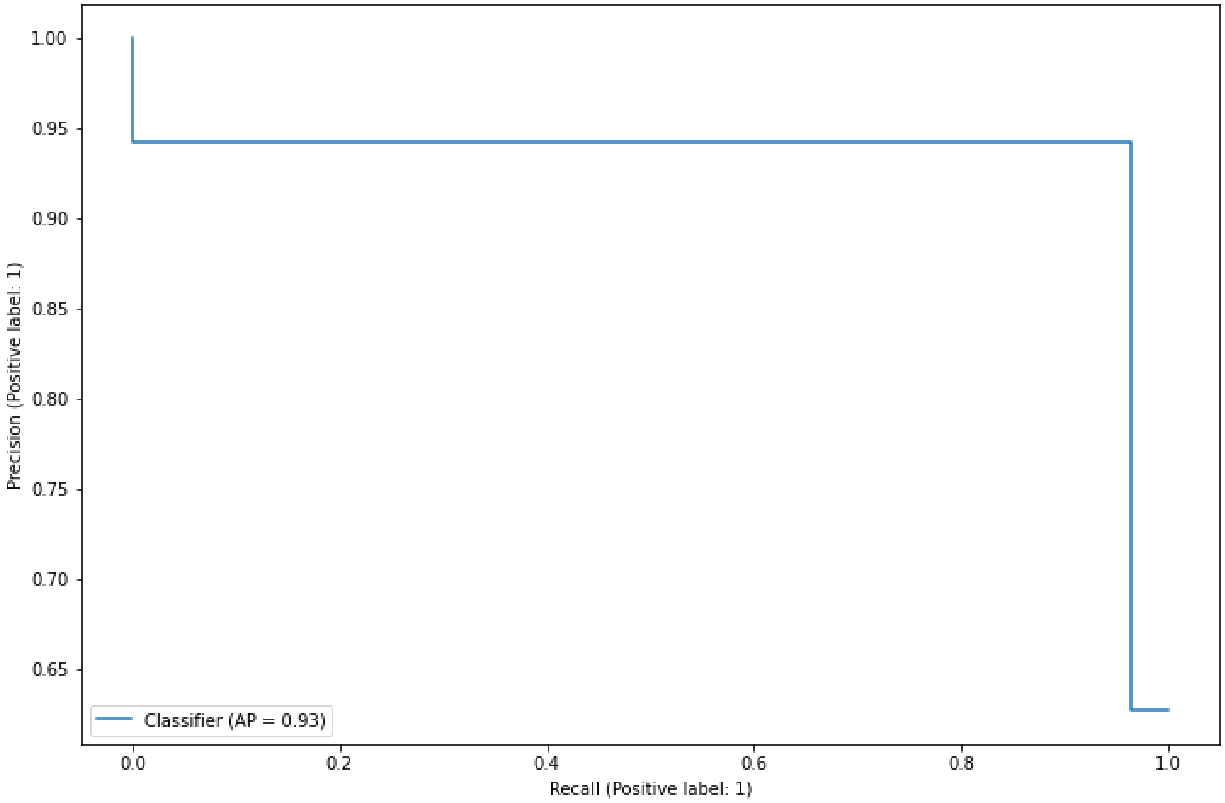
Out[34]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x21de10bd880>
```



In [35]:

```
PrecisionRecallDisplay.from_predictions(expected, predicted)
plt.show()
```


1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Precision (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)
Classifier (AP = 0.93)