Naive Bayes

=> A simple way to predict categories based on characteristics, assuming they're independent.
=> Naive bayes is simple machine learning algorithm that predicts the likelihood of something belonging
to a particular category based on its characteristics, assuming the characteristics are independent of each other.

for single variable: P(A|B) = P(B|A)*P(A)/P(B)

For multiple bariables:

$$P(A|B1,B2,B3,.....,Bn) = \frac{P(B1|A)*P(B2|A)*P(B3|A)*.....*P(B1|A)}{P(B1)*P(B2)*.....*P(Bn)}$$

Naive Bayes is of three types:

1.Gaussian Naive Bayes: Assumes continuous features follow a Gaussian Distribution
2.Multinomial Naive Bayes: Suitable for text classification with multinomial features.
3.Bernoulli Naive Bayes: Suitable for binary features.

In [42]:
```
from sklearn.naive_bayes import GaussianNB
gn = GaussianNB()
gn.fit(x_train,y_train)
```

Out[42]:
```
GaussianNB()
```

In [43]:
```
pred1 = gn.predict(x_test)
pred1
```

Out[43]:
```
array([1, 0, 0, ..., 0, 0, 0])
```

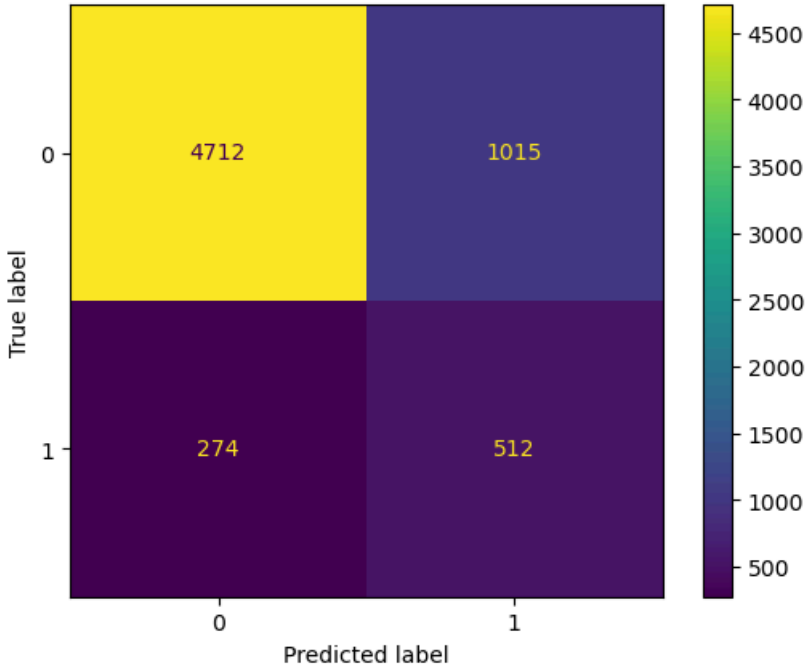
In [44]:
```
from sklearn.metrics import accuracy_score,recall_score,f1_score,precision_score
acc = accuracy_score(pred1,y_test)
recall = recall_score(pred1,y_test)
f1 = f1_score(pred1,y_test)
precision = precision_score(pred1,y_test)
print(f"Accuaracy Score : {acc}")
print(f"Recall Score : {recall}")
print(f"f1 Score : {f1}")
print(f"Precision Score : {precision}")
```

```
Accuaracy Score : 0.8020881314294488
Recall Score : 0.6513994910941476
f1 Score : 0.4427150886294855
Precision Score : 0.33529796987557303
```

In [45]:
```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
cm = confusion_matrix(pred1,y_test)
cm
```

Out[45]:
```
array([[4712, 1015],
       [ 274,  512]], dtype=int64)
```

In [46]:
```
cmd = ConfusionMatrixDisplay(cm)
cmd.plot()
plt.show()
```



KNN: K-Nearest Neighbors

KNN is a simple Machine learning algorithm that classifies a new data point based on the majority vote of its k nearest neighbors.

$$d = \sqrt{(x2-x1)^2+(y2-y1)^2}$$

In [ ]:
```
```