Naive Bayes Classifier in Machine Learning (Sklearn)

Pre-Processing Data

Memanggil data

In [1]:
```
import pandas as pd
csv_path='http://bit.ly/PHN-IRIS'
data=pd.read_csv(csv_path, index_col=0)
data.head(10)
```

Out[1]:

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa

Cek missing values

In [2]:
```
data.isna().sum()
```

Out[2]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

Cek kelas yang ada

In [3]:
```
data.loc[:,'Species'].unique()
```

Out[3]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

Encoder kolom variety karena masih dalam bentuk kategorial

In [4]:
```
from sklearn.preprocessing import LabelEncoder

label_encoder=LabelEncoder().fit(data['Species'])
```

```
data['Species']= label_encoder.transform(data['Species'])
data.head()
```

Out[4]:

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	0
2	4.9	3.0	1.4	0.2	0
3	4.7	3.2	1.3	0.2	0
4	4.6	3.1	1.5	0.2	0
5	5.0	3.6	1.4	0.2	0

Splitting Data

Menentukan data X (response) dan y (class)

Dalam Contoh Kali ini fitur X hanya menggunakan kolom 'outlook' dan 'temp'. Sedangkan untuk kelas tetap menggunakan fitur 'play'

In [5]:

```
X = data.drop(['Species'], axis=1)
Y = data.Species

X.head()
```

Out[5]:

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2

Teknik splitting data dengan Scikit-Learn

Training 80%, testing 20%

In [6]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)
```

In [7]:

```
print(X_test)
```

```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Id
147            6.3           2.5            5.0           1.9
99             5.1           2.5            3.0           1.1
4              4.6           3.1            1.5           0.2
83             5.8           2.7            3.9           1.2
20             5.1           3.8            1.5           0.3
```

3	4.7	3.2	1.3	0.2
122	5.6	2.8	4.9	2.0
52	6.4	3.2	4.5	1.5
133	6.4	2.8	5.6	2.2
49	5.3	3.7	1.5	0.2
76	6.6	3.0	4.4	1.4
10	4.9	3.1	1.5	0.1
85	5.4	3.0	4.5	1.5
90	5.5	2.5	4.0	1.3
8	5.0	3.4	1.5	0.2
112	6.4	2.7	5.3	1.9
93	5.8	2.6	4.0	1.2
116	6.4	3.2	5.3	2.3
109	6.7	2.5	5.8	1.8
140	6.9	3.1	5.4	2.1
22	5.1	3.7	1.5	0.4
73	6.3	2.5	4.9	1.5
53	6.9	3.1	4.9	1.5
100	5.7	2.8	4.1	1.3
120	6.0	2.2	5.0	1.5
89	5.6	3.0	4.1	1.3
131	7.4	2.8	6.1	1.9
32	5.4	3.4	1.5	0.4
67	5.6	3.0	4.5	1.5
135	6.1	2.6	5.6	1.4

Training Model

In [8]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB

model_NB = GaussianNB()
model_NB
```

Out[8]:
```
GaussianNB()
```

Naive bayes itu adalah $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

In [9]:
```
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
```

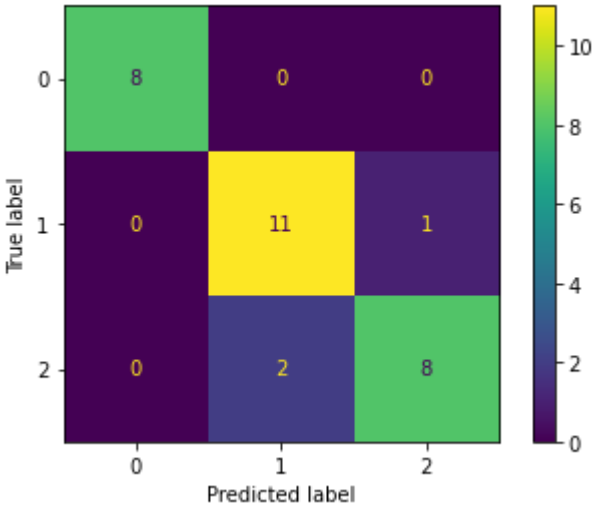
```
[2 1 0 1 0 0 2 1 2 0 1 0 1 1 0 2 1 2 2 2 0 1 2 1 1 1 2 0 1 1]
```

Evaluation

In [10]:
```
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x259fab5b370>
```

Out[10]:



In [11]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test, Y_pred_NB))
```

```
[[ 8  0  0]
 [ 0 11  1]
 [ 0  2  8]]
```

In [12]:

```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       0.85      0.92      0.88        12
           2       0.89      0.80      0.84        10

    accuracy                           0.90        30
   macro avg       0.91      0.91      0.91        30
weighted avg       0.90      0.90      0.90        30
```

Multinomial

In [13]:

```
from sklearn.naive_bayes import MultinomialNB

model_NB = MultinomialNB(alpha=1.0)
model_NB
```

Out[13]:

```
MultinomialNB()
```

In [14]:

```
import numpy as np
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
print(np.array(Y_test))
```

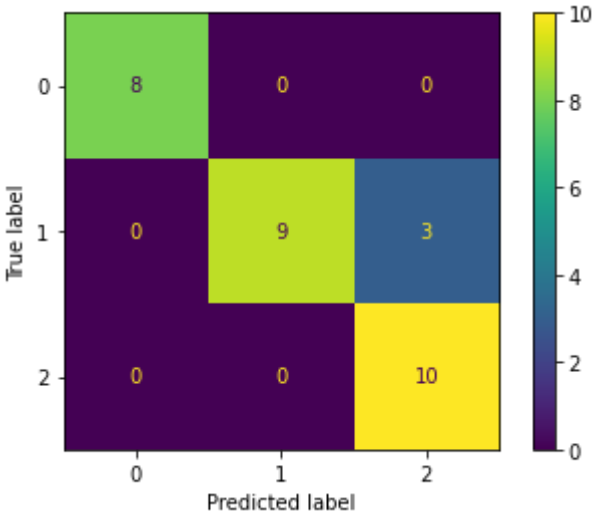
```
[2 1 0 1 0 0 2 1 2 0 1 0 2 1 0 2 1 2 2 2 0 2 1 1 2 1 2 0 2 2]
[2 1 0 1 0 0 2 1 2 0 1 0 1 1 0 2 1 2 2 2 0 1 1 1 2 1 2 0 1 2]
```

In [15]:

```
plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[15]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2598065ef40>



In [16]:
```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      0.75      0.86        12
           2       0.77      1.00      0.87        10

    accuracy                           0.90        30
   macro avg       0.92      0.92      0.91        30
weighted avg       0.92      0.90      0.90        30
```

EXERCISE/ HOMEWORK

1. Buatlah program klasifikasi diabetes menggunakan model Naive Bayes. Data dapat diunduh pada tautan berikut ini https://github.com/dhirajk100/Naive-Bayes/blob/master/Naive-Bayes-Classification-Data.csv

1. Buatlah program klasifikasi Social Media Analysis menggunakan model Naive Bayes. Data dapat diunduh pada tautan berikut ini https://www.kaggle.com/rakeshrau/social-network-ads

1. Gunakan fungsi Naive Bayes lainnya selain Gaussian dan Multinomial NB. Lihat https://scikit-learn.org/stable/modules/naive_bayes.html

1. Buatlah program klasifikasi diabetes menggunakan model Naive Bayes. Data dapat diunduh pada tautan berikut ini

https://github.com/dhirajk100/Naive-Bayes/blob/master/Naive-Bayes-Classification-Data.csv

Pre-Processing Data

Memanggil data

In [17]:
```
import pandas as pd
data=pd.read_csv('Naive-Bayes-Classification-Data.csv')
data
```

Out[17]:

	glucose	bloodpressure	diabetes
0	40	85	0
1	40	92	0
2	45	63	1
3	45	80	0
4	40	73	1
...	...	...	...
990	45	87	0
991	40	83	0
992	40	83	0
993	40	60	1
994	45	82	0

995 rows × 3 columns

Cek missing values

In [18]:
```
data.isna().sum()
```

Out[18]:
```
glucose          0
bloodpressure    0
diabetes         0
dtype: int64
```

In [19]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 995 entries, 0 to 994
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   glucose        995 non-null    int64
 1   bloodpressure  995 non-null    int64
 2   diabetes       995 non-null    int64
dtypes: int64(3)
memory usage: 23.4 KB
```

Cek kelas yang ada

In [20]:
```
data.loc[:,'diabetes'].unique()
```

```
array([0, 1], dtype=int64)
```

Out[20]:

Splitting Data

Menentukan data X (response) dan y (class)

In [21]:
```
X = data.drop(['diabetes'], axis=1)
Y = data.diabetes

X.head()
```

Out[21]:

	glucose	bloodpressure
0	40	85
1	40	92
2	45	63
3	45	80
4	40	73

Teknik splitting data dengan Scikit-Learn

Training 80%, testing 20%

In [22]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)
```

In [23]:
```
print(X_test)
```

```
     glucose  bloodpressure
438       55             70
836       40             85
608       45             73
759       60             75
453       45             90
..       ...            ...
146       50             77
446       50             90
589       50             83
756       55             67
762       40             93

[199 rows x 2 columns]
```

Training Model

In [24]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB

model_NB = GaussianNB()
model_NB
```

Out[24]:
```
GaussianNB()
```

Naive bayes itu adalah $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

In [25]:
```
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
```

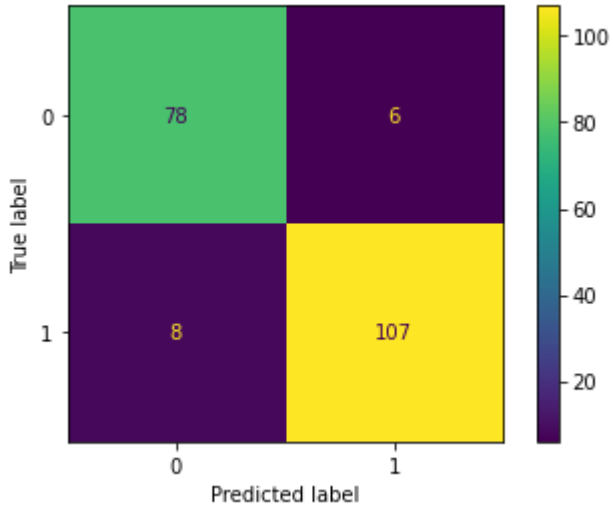
```
[1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0
 0 1 1 1 1 1 1 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0
 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1 1
 1 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0
 0 0 0 1 1 1 1 1 1 1 0 0 1 0]
```

Evaluation

In [26]:
```
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[26]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x259807d9490>
```



In [27]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test, Y_pred_NB))
```

```
[[ 78   6]
 [  8 107]]
```

In [28]:
```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support

           0       0.91      0.93      0.92        84
           1       0.95      0.93      0.94       115

    accuracy                           0.93       199
   macro avg       0.93      0.93      0.93       199
weighted avg       0.93      0.93      0.93       199
```

ComplementNB

In [29]:
```
from sklearn.naive_bayes import ComplementNB

model_NB = ComplementNB()
model_NB
```

Out[29]:
```
ComplementNB()
```

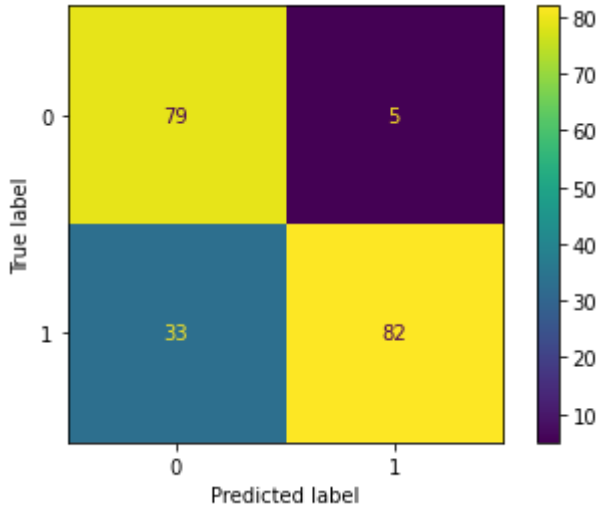
In [30]:
```
import numpy as np
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
print(np.array(Y_test))
```

```
[1 0 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 0 0 1 1 0
 1 1 1 1 0 1 0 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1
 1 1 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0
 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0
 1 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0
 0 0 0 1 1 1 1 0 1 1 0 1 1 0]
[1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0
 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0
 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1
 1 0 1 0 1 1 1 1 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0
 0 0 0 1 1 1 1 1 1 1 0 0 1 0]
```

In [31]:
```
plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[31]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x259fcf10190>
```



In [32]:
```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support
```

```
              0       0.71      0.94      0.81        84
              1       0.94      0.71      0.81       115

    accuracy                           0.81       199
   macro avg       0.82      0.83      0.81       199
weighted avg       0.84      0.81      0.81       199
```

2. Buatlah program klasifikasi Social Media Analysis menggunakan model Naive Bayes. Data dapat diunduh pada tautan berikut ini https://www.kaggle.com/rakeshrau/social-network-ads

Pre-Processing Data

Memanggil data

In [33]:
```
import pandas as pd
data=pd.read_csv('Social_Network_Ads.csv', index_col=0)
data.head(10)
```

Out[33]:

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0

Cek missing values

In [34]:
```
data.isna().sum()
```

Out[34]:
```
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [35]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 400 entries, 15624510 to 15594041
```

```
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           400 non-null    object
 1   Age              400 non-null    int64
 2   EstimatedSalary  400 non-null    int64
 3   Purchased        400 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.6+ KB
```

Cek kelas yang ada

In [36]:
```
data.loc[:,'Purchased'].unique()
```

Out[36]:
```
array([0, 1], dtype=int64)
```

Encoder kolom gender karena masih dalam bentuk object

In [37]:
```
from sklearn.preprocessing import LabelEncoder

label_encoder=LabelEncoder().fit(data['Gender'])

data['Gender']= label_encoder.transform(data['Gender'])
data.head()
```

Out[37]:

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

Splitting Data

Menentukan data X (response) dan y (class)

In [38]:
```
X = data.drop(['Purchased'], axis=1)
Y = data.Purchased

X.head()
```

Out[38]:

User ID	Gender	Age	EstimatedSalary
15624510	1	19	19000
15810944	1	35	20000
15668575	0	26	43000
15603246	0	27	57000
15804002	1	19	76000

Teknik splitting data dengan Scikit-Learn

Training 80%, testing 20%

In [39]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)
```

In [40]:
```
print(X_test)
```

```
          Gender  Age  EstimatedSalary
User ID
15794493       1   40            57000
15570932       1   34           115000
15571059       0   33            41000
15697997       0   38            80000
15741049       1   37            72000
...          ...  ...              ...
15591433       1   36            52000
15627220       1   39            71000
15768072       0   47            50000
15594577       1   25            22000
15669656       1   31            18000

[80 rows x 3 columns]
```

Training Model

In [41]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB

model_NB = GaussianNB()
model_NB
```

Out[41]:
```
GaussianNB()
```

Naive bayes itu adalah $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

In [42]:
```
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
```

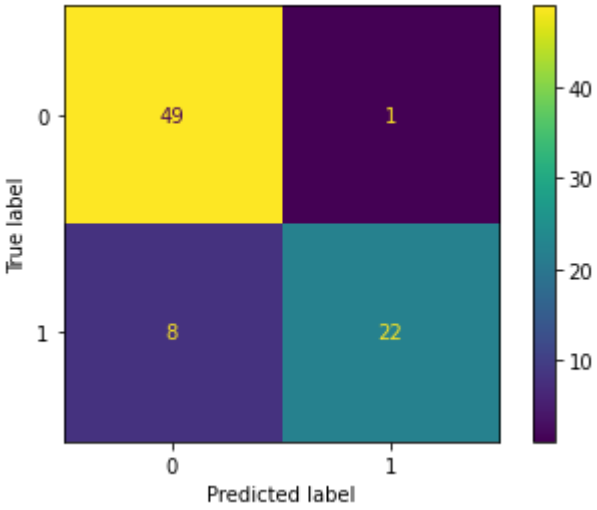
```
[0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 1 0 0 0 1 1 1 0 0 0 0
 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0
 1 0 0 0 0 0]
```

Evaluation

In [43]:
```
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[43]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x25980939400>



In [44]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test, Y_pred_NB))
```

```
[[49  1]
 [ 8 22]]
```

In [45]:
```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support

           0       0.86      0.98      0.92        50
           1       0.96      0.73      0.83        30

    accuracy                           0.89        80
   macro avg       0.91      0.86      0.87        80
weighted avg       0.90      0.89      0.88        80
```

ComplementNB

In [46]:
```
from sklearn.naive_bayes import ComplementNB
model_NB = ComplementNB()
model_NB
```

Out[46]: ComplementNB()

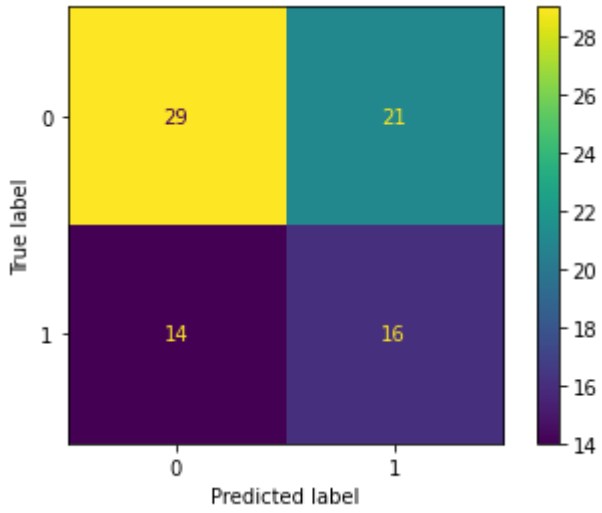
In [47]:
```
import numpy as np
model_NB.fit(X_train,Y_train)

Y_pred_NB=model_NB.predict(X_test)

print(Y_pred_NB)
print(np.array(Y_test))
```

```
[0 1 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 0 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 1 1
 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1
 0 0 0 0 0 0]
[0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0
 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 1
 1 0 0 1 0 0]
```

In [48]:
```
plot_confusion_matrix(model_NB, X_test, Y_test)
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[48]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x259807d98b0>
```



In [49]:
```
print(classification_report(Y_test, Y_pred_NB))
```

```
              precision    recall  f1-score   support

           0       0.67      0.58      0.62        50
           1       0.43      0.53      0.48        30

    accuracy                           0.56        80
   macro avg       0.55      0.56      0.55        80
weighted avg       0.58      0.56      0.57        80
```