

In []: *#full code in 1 cell*

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

df=pd.read_csv('iris (3).csv')

df.isnull().sum()

df.head()

X=df[['sepal_length','sepal_width','petal_length','petal_width']]
y=df['species']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)

model=GaussianNB()
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

cm=confusion_matrix(y_test,y_pred)
print(cm)

accuracy=accuracy_score(y_test,y_pred)
print(accuracy)

error=1-accuracy
print(error)

precision=precision_score(y_test,y_pred,average='macro')
print(precision)

recall=recall_score(y_test,y_pred,average='macro')
print(recall)
```

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.naive_bayes import GaussianNB
        from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
```

```
In [3]: df=pd.read_csv('iris (3).csv')
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: sepal_length    0
        sepal_width     0
        petal_length    0
        petal_width     0
        species         0
        dtype: int64
```

```
In [5]: df.head()
```

```
Out[5]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	6.2	3.4	5.4	2.3	Iris-virginica
3	5.9	3.0	5.1	1.8	Iris-virginica
4	6.0	2.2	4.0	1.0	Iris-versicolor

```
In [8]: X=df[['sepal_length','sepal_width','petal_length','petal_width']]
        y=df['species']
```

```
In [9]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)
```

```
In [10]: model=GaussianNB()
         model.fit(X_train,y_train)
```

Out[10]:

▼ GaussianNB ⓘ ?

GaussianNB()

In [13]: `y_pred=model.predict(X_test)`In [22]: `cm=confusion_matrix(y_test,y_pred)`
`print(cm)`

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

In [23]: `accuracy=accuracy_score(y_test,y_pred)`
`print(accuracy)`

0.6666666666666666

In [25]: `precision=precision_score(y_test,y_pred,average='macro')`
`print(precision)`

0.5

c:\users\chatura karankal\appdata\local\programs\python\python39\lib\site-packages\sklearn\metrics_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

In [27]: `recall=recall_score(y_test,y_pred,average='macro')`
`print(recall)`

0.6666666666666666

In [28]: `error=1-accuracy`
`print(error)`

0.3333333333333333

In []: