

```
In [ ]: NAME : SHINDE SHUBHAM DNYANDEV, ROLL NO. : EN23107121, BATCH : C
```

```
In [1]: import pandas as pd
```

```
In [3]: df = pd.read_csv("/home/admin1/iris.csv")
df
```

```
Out[3]:      sepal_length  sepal_width  petal_length  petal_width  species
 0           5.1          3.5          1.4          0.2    setosa
 1           4.9          3.0          1.4          0.2    setosa
 2           4.7          3.2          1.3          0.2    setosa
 3           4.6          3.1          1.5          0.2    setosa
 4           5.0          3.6          1.4          0.2    setosa
 ...
 145          6.7          3.0          5.2          2.3  virginica
 146          6.3          2.5          5.0          1.9  virginica
 147          6.5          3.0          5.2          2.0  virginica
 148          6.2          3.4          5.4          2.3  virginica
 149          5.9          3.0          5.1          1.8  virginica
```

150 rows × 5 columns

```
In [9]: df.describe()
```

```
Out[9]:      sepal_length  sepal_width  petal_length  petal_width
count    150.000000    150.000000    150.000000    150.000000
mean     5.843333     3.054000     3.758667     1.198667
std      0.828066     0.433594     1.764420     0.763161
min      4.300000     2.000000     1.000000     0.100000
25%     5.100000     2.800000     1.600000     0.300000
50%     5.800000     3.000000     4.350000     1.300000
75%     6.400000     3.300000     5.100000     1.800000
max     7.900000     4.400000     6.900000     2.500000
```

```
In [11]: df.isnull()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [13]: `df.isnull().sum()`

Out[13]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [15]: `df.notnull()`

Out[15]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	True	True	True	True	True
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	True	True	True
...	...	...	...	...	...
145	True	True	True	True	True
146	True	True	True	True	True
147	True	True	True	True	True
148	True	True	True	True	True
149	True	True	True	True	True

150 rows × 5 columns

In [17]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   sepal_length  150 non-null   float64 
 1   sepal_width   150 non-null   float64 
 2   petal_length  150 non-null   float64 
 3   petal_width   150 non-null   float64 
 4   species       150 non-null   object  
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
In [29]: x = df.drop(["species"], axis = 1)
y = df["species"]
```

```
In [31]: x
```

```
Out[31]:    sepal_length  sepal_width  petal_length  petal_width
0           5.1          3.5          1.4          0.2
1           4.9          3.0          1.4          0.2
2           4.7          3.2          1.3          0.2
3           4.6          3.1          1.5          0.2
4           5.0          3.6          1.4          0.2
...
145          6.7          3.0          5.2          2.3
146          6.3          2.5          5.0          1.9
147          6.5          3.0          5.2          2.0
148          6.2          3.4          5.4          2.3
149          5.9          3.0          5.1          1.8
```

150 rows × 4 columns

```
In [33]: y
```

```
Out[33]: 0      setosa
1      setosa
2      setosa
3      setosa
4      setosa
...
145     virginica
146     virginica
147     virginica
148     virginica
149     virginica
Name: species, Length: 150, dtype: object
```

```
In [27]: from sklearn.model_selection import train_test_split
```

```
In [65]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

```
In [67]: from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
```

```
In [69]: GNB = GaussianNB()
GNB
```

```
Out[69]: ▾ GaussianNB
          GaussianNB()

In [71]: GNB.fit(x_train, y_train)

Out[71]: ▾ GaussianNB
          GaussianNB()

In [73]: y_pred = GNB.predict(x_test)
          y_pred

Out[73]: array(['versicolor', 'virginica', 'setosa', 'virginica', 'versicolor',
       'virginica', 'versicolor', 'versicolor', 'setosa', 'versicolor',
       'setosa', 'setosa', 'virginica', 'setosa', 'virginica',
       'versicolor', 'virginica', 'versicolor', 'setosa', 'virginica',
       'versicolor', 'setosa', 'setosa', 'virginica', 'virginica',
       'setosa', 'versicolor', 'virginica', 'virginica', 'virginica'],
      dtype='|<U10')

In [75]: from sklearn.metrics import confusion_matrix

In [77]: CM = confusion_matrix(y_test, y_pred)
          CM

Out[77]: array([[ 9,  0,  0],
       [ 0,  9,  1],
       [ 0,  0, 11]])

In [79]: from sklearn.metrics import accuracy_score

In [81]: AS = accuracy_score(y_test, y_pred)
          AS

Out[81]: 0.9666666666666667

In [84]: from sklearn.metrics import classification_report

In [86]: CR = classification_report(y_test, y_pred)
          CR

Out[86]: 'precision    recall   f1-score   support\n\nversicolor      1.00      0.90      0.95      10\nvirginica       0.92      1.00      0.96      11\nmacro avg       0.97      0.97      0.97      30\nweighted avg    0.97      0.97      0.97      30'

In [88]: MNB = MultinomialNB()
          MNB

Out[88]: ▾ MultinomialNB
          MultinomialNB()

In [90]: MNB.fit(x_train, y_train)

Out[90]: ▾ MultinomialNB
          MultinomialNB()
```

```
In [92]: y_pred = MNB.predict(x_test)
y_pred
```

```
Out[92]: array(['versicolor', 'virginica', 'setosa', 'virginica', 'versicolor',
   'virginica', 'versicolor', 'versicolor', 'setosa', 'versicolor',
   'setosa', 'setosa', 'virginica', 'setosa', 'virginica',
   'versicolor', 'virginica', 'versicolor', 'setosa', 'virginica',
   'versicolor', 'setosa', 'setosa', 'virginica', 'virginica',
   'setosa', 'versicolor', 'virginica', 'virginica', 'versicolor'],
  dtype='|<U10')
```

```
In [94]: from sklearn.metrics import confusion_matrix
```

```
In [96]: CM = confusion_matrix(y_test, y_pred)
CM
```

```
Out[96]: array([[ 9,  0,  0],
   [ 0,  9,  1],
   [ 0,  1, 10]])
```

```
In [98]: from sklearn.metrics import accuracy_score
```

```
In [100]: AS = accuracy_score(y_test, y_pred)
AS
```

```
Out[100]: 0.9333333333333333
```

```
In [102]: from sklearn.metrics import classification_report
```

```
In [104]: CR = classification_report(y_test, y_pred)
CR
```

```
Out[104]:          precision    recall  f1-score   support\n          setosa      1.00      1.00      1.00      15\n versicolor      0.91      0.91      0.91      14\n virginica      0.93      0.93      0.93      12\n          avg /宏平均      0.93      0.93      0.93      41\n          avg /加权平均      0.93      0.93      0.93      41
```

```
In [106]: BNB = BernoulliNB()
BNB
```

```
Out[106]: ▾ BernoulliNB
BernoulliNB()
```

```
In [108]: BNB.fit(x_train, y_train)
```

```
Out[108]: ▾ BernoulliNB
BernoulliNB()
```

```
In [124]: y_pred = BNB.predict(x_test)
y_pred
```

```
Out[124]: array(['setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
   'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
   'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
   'setosa', 'setosa', 'setosa', 'setosa', 'setosa'],
  dtype='|<U10')
```

```
In [126]: from sklearn.metrics import confusion_matrix
```

```
In [128]: CM = confusion_matrix(y_test, y_pred)
CM
```

```
Out[128]: array([[ 9,  0,  0],
   [10,  0,  0],
   [11,  0,  0]])
```

```
In [130]: from sklearn.metrics import accuracy_score
```

```
In [132]: AS = accuracy_score(y_test, y_pred)
AS
```

```
Out[132]: 0.3
```

```
In [134]: from sklearn.metrics import classification_report
```

```
In [140]: CR = classification_report(y_test, y_pred)
CR
```

```
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
```

```
Out[140]:          precision    recall  f1-score   support\n\n        1.00      0.46      9\nversicolor      0.00      0.00      0.00       0.00      0.00      0.00\nvirginica      0.30      30\nmacro avg      0.10      0.10      0.10      0.33      0.15      0.30\nweighted avg   0.09      0.30      0.14      30\n'
```

```
In [ ]:
```