```
from google.colab import files
uploaded_files = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving IRIS.csv to IRIS.csv
```

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('IRIS.csv')
```

```
df
```

	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	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

```
df.isnull().sum()
```

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

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
df['species'] = le.fit_transform(df['species'])
```

```
df['species']
```

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 150, dtype: int64
```

```
x = df.iloc[:,0:4]
```

```
x
```

	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

```
y = df.iloc[:,-1]
```

```
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 150, dtype: int64
```

```
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)
```

```
x_train
```

	sepal_length	sepal_width	petal_length	petal_width
37	4.9	3.1	1.5	0.1
100	6.3	3.3	6.0	2.5
56	6.3	3.3	4.7	1.6
81	5.5	2.4	3.7	1.0
145	6.7	3.0	5.2	2.3
...	...	...	...	...
147	6.5	3.0	5.2	2.0
141	6.9	3.1	5.1	2.3
82	5.8	2.7	3.9	1.2
111	6.4	2.7	5.3	1.9
116	6.5	3.0	5.5	1.8

120 rows × 4 columns

```
x_test
```

	sepal_length	sepal_width	petal_length	petal_width
7	5.0	3.4	1.5	0.2
124	6.7	3.3	5.7	2.1
89	5.5	2.5	4.0	1.3
14	5.8	4.0	1.2	0.2
103	6.3	2.9	5.6	1.8
51	6.4	3.2	4.5	1.5
132	6.4	2.8	5.6	2.2
92	5.8	2.6	4.0	1.2
23	5.1	3.3	1.7	0.5
90	5.5	2.6	4.4	1.2
102	7.1	3.0	5.9	2.1
69	5.6	2.5	3.9	1.1
70	5.9	3.2	4.8	1.8
8	4.4	2.9	1.4	0.2
59	5.2	2.7	3.9	1.4
40	5.0	3.5	1.3	0.3
24	4.8	3.4	1.9	0.2
18	5.7	3.8	1.7	0.3
99	5.7	2.8	4.1	1.3
79	5.7	2.6	3.5	1.0

```
y_train
```

```
37     0
100    2
56     1
81     1
145    2
      ..
147    2
141    2
82     1
111    2
116    2
Name: species, Length: 120, dtype: int64
```

```
y_test
```

```
7      0
124    2
89     1
14     0
103    2
51     1
132    2
92     1
23     0
90     1
102    2
69     1
70     1
8      0
59     1
40     0
24     0
18     0
99     1
79     1
72     1
143    2
128    2
60     1
109    2
26     0
80     1
138    2
4      0
28     0
Name: species, dtype: int64
```

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
```

```
model = nb.fit(x_train,y_train)
```

```
model
```

```
▾ GaussianNB
GaussianNB()
```

```
y_pred = model.predict(x_test)
```

```
y_pred
```

```
array([0, 2, 1, 0, 2, 1, 2, 1, 0, 1, 2, 1, 2, 0, 1, 0, 0, 0, 1, 1, 1, 2,
       2, 1, 2, 0, 1, 2, 0, 0])
```

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
```

```
mse
```

```
0.03333333333333333
```

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test,y_pred)
```

```
accuracy
```

```
0.9666666666666667
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
```

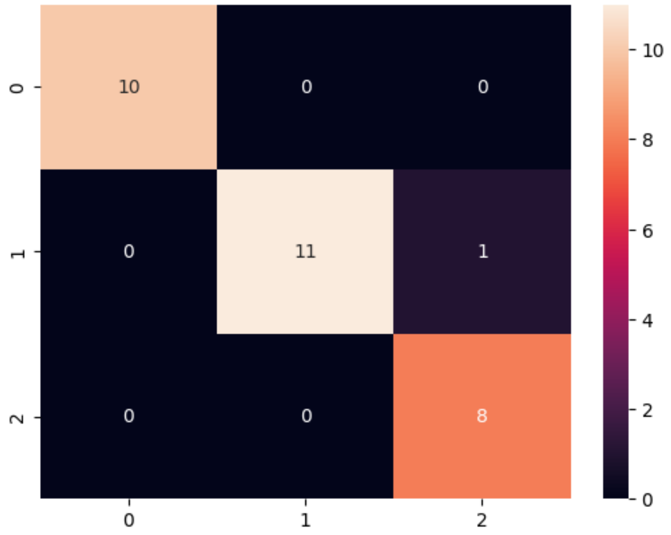
```
cm
```

```
array([[10,  0,  0],
       [ 0, 11,  1],
       [ 0,  0,  8]])
```

```
import seaborn as sns
```

```
sns.heatmap(cm,annot=True)
```

```
<Axes: >
```



```
from sklearn.metrics import precision_score
ps = precision_score(y_test,y_pred,average="micro")
```

```
ps
```

```
0.9666666666666667
```

```
from sklearn.metrics import recall_score
rs = recall_score(y_test,y_pred,average="micro")
```

```
rs
```

```
0.9666666666666667
```

✓ 0s completed at 6:22 PM ● ✕